Karlsruhe Institute of Technology

Analysis of Heuristics for Treewidth

Bachelor Thesis of

Marcus Wunderlich

At the Department of Informatics
Institute of Theoretical Informatics

Reviewers:	TT.-Prof. Dr. Thomas Bläsius
	PD Dr. Torsten Ueckerdt
Advisors:	Marcus Wilhelm

Time Period: 20.5.2021 – 20.9.2021

Statement of Authorship

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, September 16, 2021

Abstract

Treewidth is an important concept in many fields like for example, graph theory and parameterized algorithms. Determining a graph's treewidth is necessary for many applications, but it is, unfortunately, an $\mathcal{NP}$-hard problem. To at least approximate the treewidth efficiently, several heuristics were introduced, including the MinDegree heuristic and the MinFillIn heuristic. Although these achieve surprisingly good results in practice, they are barely studied theoretically. In this thesis, we initiate this study by investigating the behavior of the two heuristics on grids. Along the way to grids with arbitrarily large side lengths, we also investigate some superclasses of small grids like the chordal graphs and series-parallel graphs. We additionally present a first approach to upper bound the result of the MinDegree heuristic, and introduce the concept of a border graph; a simple structure that captures all relevant information at any fixed point in time during the execution of one of the heuristics. We believe that the border graph provides a useful perspective that will help future studies of the heuristics.

Zusammenfassung

Das Konzept der Baumweite spielt in vielen Bereichen, wie der parametrisierten Algorithmik und der Graphentheorie, eine wichtige Rolle. Es ist zwar für viele Anwendungen notwendig, die Baumweite eines Graphen zu bestimmen, allerdings ist das Problem für allgemeine Graphen $\mathcal{NP}$-schwer. Um die Baumweite eines Graphens wenigstens zu approximieren, wurde verschiedenste Heuristiken entwickelt, u.a. die MinDegree- und die MinFillIn-Heuristik. Obwohl beide Heuristiken überraschend gute Ergebnisse in der Praxis erzielen, wurde sie theoretisch bisher kaum untersucht. In dieser Arbeit beginnen wir diese Untersuchung, indem wir uns auf das Verhalten dieser Heuristiken auf Gittern konzentrieren. Im Laufe der Arbeit untersuchen wir Gitter mit größer werdenden Seitenlängen, und beschreiben dabei auch das Verhalten der Heuristiken auf manchen Superklassen kleinerer Gitter, wie zum Beispiel den chordalen Graphen und den seriell-parallelen Graphen. Außerdem stellen wir eine Methode vor, mit der eine obere Schranke für das Ergebnis der Heuristiken auf Graphen gefunden werden kann, und führen die „border graphs“ ein; eine Graph-Art, die alle relevanten Informationen während des Ablaufes der Heuristiken kompakt darstellt. Wir glauben, dass das Konzept der „border graphs“ hilfreich für weitere Arbeiten in diesem Gebiet sein wird.

Contents

1. Introduction

It is well known that some problems are, although being $\mathcal{NP}$-hard on arbitrary graphs, efficiently solvable on special graph classes. The field of parameterized algorithms studies for which graph classes this applies by studying the parameters that are responsible for the exponential running time. If such a parameter can be bounded from above for certain instances, this directly transfers to an upper bound to the running time of the algorithms. One of the most used parameters is the treewidth of a graph, which indicates the similarity of a graph to a tree. The idea of this graph parameter is that trees are algorithmically easy to handle, and that graphs similar to trees might be manageable as well. In fact, there exist polynomial algorithms for many generally $\mathcal{NP}$-hard problems on graphs with bounded treewidth. However, they need to know the treewidth of a graph in advance, and determining the treewidth of a graph is again $\mathcal{NP}$-hard in general.

In the Parameterized Algorithms and Computational Experiments (PACE) Challenge, several exact algorithms were proposed that perform well in practice [DHJ+16]. A crucial part of these algorithms are heuristics to approximate the treewidth of a graph, some of which are the MinDegree and MinFillIn heuristic. Besides its importance for the determination and approximation of treewidth, the MinDegree heuristic is a key part in the field of sparse matrix computation and its applications.

The theoretical basis for the heuristics is the fact that the treewidth of a graph is the treewidth of a smallest chordal supergraph. The chordal graphs are those graphs with a perfect elimination scheme, i.e., a vertex ordering such that for each vertex the neighbors with a higher number form a clique. Their treewidth can be easily determined by finding the size of a maximal clique, which is possible in linear time on chordal graphs thanks to their perfect elimination scheme. The heuristics guess an elimination scheme and make it perfect by adding missing edges if the higher numbered neighborhood of a vertex is not a clique. The number of edges added that way is the fill-in of a vertex, which is the namesake for the MinFillIn heuristic. The idea of this heuristic to find a minimal chordal supergraph is to add as few as possible edges. The MinDegree heuristic, on the other hand, tries to build only small cliques to keep the treewidth of the chordal supergraph as small as possible.

Both, the MinDegree as well as the MinFillIn heuristic, achieve results that are surprisingly close to the instance's actual treewidth in practice, despite being much faster than exact algorithms. However, so far there are only a few theoretical investigations on why they perform so well.

A good starting point for these investigations is the behavior of the heuristics on grids. The class of grid graphs is closely linked to the concept of treewidth due to the excluded grid theorem presented by Robertson and Seymour in 1986 [RS86b]. It states that every graph with sufficient large treewidth has an $n \times n$-grid as minor. The proof of this theorem shows that such a graph contains a series of increasingly ordered structures, the last of which is a grid. These structures can serve as orientation points in the study of the heuristics behavior from grids to general graphs.

1.1 Contribution and Outline

In this thesis, we contribute to the theoretical study of the MinDegree and the MinFillIn heuristic by investigating their behavior on grids. After introducing necessary notations and concepts in Chapter 2, we construct graph families on which the MinDegree heuristic returns better approximations than the MinFillIn heuristic, and the other way round. Then, in the course of this thesis, we consider grids with increasing side lengths, starting with sides lengths of at most 3 for the MinFillIn and MinDegree heuristic in Chapters 4 and 5 respectively. In Chapter 4 we show that the MinFillIn heuristic returns the correct treewidth for all chordal graphs, some graphs of treewidth 2 including $2 \times n$-grids, and $3 \times n$-grids. For the MinDegree heuristic we show the same statement for even more graphs with treewidth at most 3. We, additionally, transfer our findings to the next most ordered structure, that is used in the proof of the excluded grid theorem. In Chapter 6 we present two approaches to find an upper bound to the result of the MinDegree heuristic on larger grids. One approach yields the quadratic upper bound $3/4(n^2 - n)$, where n is the side length of a quadratic grid. The other one is about how the grid's structure changes during the heuristic. In this section, we introduce several concepts and ideas, of which we think, are helpful for further investigation on these heuristics. Finally, we review our findings and give remarks about open questions in Chapter 7.

1.2 Related Work

The term treewidth was introduced by Robertson and Seymour in 1986 [RS86a] via tree decompositions. Over time many new equivalent definitions and characterizations were introduced using for example minimal triangulations and perfect elimination orderings [PS97] or brambles [ST93]. An overview over most characterizations was written by Bodlaender [Bod98]. Since most characterizations are formulated as optimization problems, they often imply heuristics to find valid but not necessarily optimal structures that have to be optimized. For example, the MinDegree and MinFillIn heuristic both find valid elimination orderings, just like many other heuristics that are based on the elimination game introduced by Bodlaender and Koster [BK10]. Recently, some improvements of these rather simple heuristics were proposed by Bachoore and Bodlaender [BB05], Clautiaux et al. [CCMN03] and Berry, Heggernes and Simonet [BHS03]. A survey over further heuristics and exact algorithms to determine the treewidth of a graph was written by Bodlander [Bod05]. For the mentioned heuristics MinDegree and MinFillIn experimental studies showed that both heuristics are often close to the actual treewidth, while the MinFillIn heuristic is often slightly better [vDvdHS06] [MSJ19a]. Theoretically, it was shown by Berry et al. that the MinDegree heuristic is robust in the sense that the chance of only adding edges that belong to a minimal triangulation remains intact even after one undesired edge was added [BHS03].

2. Preliminaries

This chapter provides the basic concepts, notations, and lemmas including treewidth with several characterizations and the heuristics we analyze in this thesis.

2.1 Basic Concepts

Graphs

An *(undirected) graph* G is a pair (V, E) of a finite set V and a set E of two-element sets of V. We call the elements of V *vertices* and the elements in E *edges*. Note that, according to the definition, neither multiple edges nor loops are allowed. Two vertices $v_1, v_2 \in V$ are called *adjacent* if there is an edge $\{v_1, v_2\} = e \in E$ in the graph. In this case v_1 and v_2 are *incident* to the edge e respectively. An edge $e = \{u, w\} \in E$ with $u \in U \subseteq V$ and $w \in W \subseteq V$ is called a UW–edge (or WU–edge). If $U = W$ we write U–edge instead of UU–edge. The *neighborhood* $N_G(v)$ of a vertex $v \in V$ is the set of all vertices that are adjacent to v in G. The number of those vertices is called the *degree of* v and is denoted as $\deg(v)$. If $\deg(v) = 1$ for some vertex v, it is called a *leaf*. The *minimal degree* of a vertex in G is denoted as $\delta(G)$. A graph in which all vertices are pairwise adjacent is called a *clique*. A clique with $n \in \mathbb{N}$ vertices is denoted as K_n or n-clique. The size of the maximal clique in a graph G is called the *clique number* $\omega(G)$ of G.

Operations on Graphs

If $G = (V, E)$ is a graph then each subset $W \subseteq V$ of its vertices *induces* a subgraph $G[W] \coloneqq (W, \{\{u, v\} \in E \mid u, v \in W\})$. As a shorthand, we write $G - W$ for the graph G without the vertices in $W \subseteq V$ and its incident edges, i.e. $G - W \coloneqq G[V \setminus W]$. If $w \in W$ is the only element in W, we write $G - w$ for $G - \{w\}$. We sometimes say that a set of vertices $W \subseteq V$ is a clique, if $G[W]$ is a clique. The operation in which all necessary edges are added to $G[W]$ such that W induces a clique in G is called a *fill-in* for W. The so created graph is denoted as $G + \text{clique}(W)$, where $\text{clique}(W)$ is the set of all missing edges in $G[W]$. The *fill-in* of a vertex $v \in V$ is the the number $|clique(N_G(v))|$.

An *edge contraction* is an operation on a graph $G = (V, E)$ in which two adjacent vertices $u, v \in V$ and the edge $\{u, v\} \in E$ are replaced by a new vertex v^* which is adjacent to all remaining vertices in the neighborhoods of u and v. Now a *minor* of a graph $G = (V, E)$ is a graph that can be obtained from G by applying a sequence of the following operations: removing a single edge, removing a vertex and its incident edges, contracting an edge. If G' is a minor of G, we write $G' \preccurlyeq G$. Note that $\preccurlyeq$ is a partial ordering of graphs.

A *subdivision* of an edge $\{v, u\}$ is an operation in which a new vertex w is added and the edge $\{v, u\}$ is replaced by the two edges $\{v, w\}$ and $\{w, u\}$. If a graph G' can be constructed from a graph G by a sequence of edge subdivisions, G' is called a *subdivision* of G.

Paths and Connections

A *path* in a graph $G = (V, E)$ is a sequence of vertices in which all directly consecutive vertices are adjacent. The first vertex of a path is called the *start vertex* and the last vertex in a path is its *end vertex*. A *cycle* is a path with the same start and end vertex. Two vertices u, v are *connected* if there is a path with u as start vertex and v as end vertex. A graph in which all its vertices are pairwise connected is called *connected* itself. In case a graph G is not connected all non-empty maximally connected subgraphs of G are referred to as *(connected) components* of G. A graph G is said to be *k-(vertex-)connected* if it has at least k vertices and it remains connected after removing less than k vertices.

Planarity

A graph is called *planar* if it can be drawn such that its edges only intersect at common vertices. If $\mathcal{D}$ is the set of points covered by this drawing, then the connected areas of $\mathbb{R}^2 \setminus \mathcal{D}$ are the *faces* of the drawing (of the graph). The only infinite face is called the *outer face*, while all other faces are *inner faces*. If all vertices of a graph can be drawn such that they border to the outer face, this graph is called *outerplanar*. Note that the set of planar graphs are closed under taking the minor [Wag37] and building subdivisions [Kur30] each. A general planar drawing can be described by giving the clockwise order of neighbors for each vertex.

2.2 Treewidth

A connected graph without cycles is called a *tree*. A *tree decomposition* of an arbitrary graph $G = (V, E)$ is a tree $T = (B, F)$, with *bags* $B \subseteq 2^V$ as vertices, that holds the following conditions.

- $\bigcup_{X \in B} X = V$
- for each edge $e = \{u, v\}$ in G there is a bag $X \in B$ with $u, v \in X$
- for each vertex $v \in V$ the bags $\{X \in B \mid v \in X\}$ induce a tree in T

The *width* of a tree decomposition $T = (B, F)$ is defined as the size of the largest bag minus 1, i.e. $\max_{X \in B} |X| - 1$. The *treewidth* of a graph is the minimum width over all its tree decompositions. We denote the treewidth of a graph G as $\mathrm{tw}(G)$. All graphs with treewidth of at most $k \in \mathbb{N}$ are called the *partial k-trees*. In this thesis, we only consider connected graphs with at least one edge. Therefore a treewidth of 0 can not occur.

2.3 Bounds for Treewidth

There are several lower and upper bounds to the treewidth of a graph arising from many different characterizations of treewidth. Most of those characterizations are defined as optimization problems. For example, the width of each arbitrary tree decomposition of a graph G is an upper bound to its treewidth $\mathrm{tw}(G)$. A simple lower bound is the clique number of the graph minus one, which can be inferred from the following Lemma 2.1.

Lemma 2.1 ([Bod05, Lemma 1])**.** *Let $W \subseteq V$ be a set of vertices of a graph $G = (V, E)$ that induces a clique in G. Then there is a bag $X \in B$ with $W \subseteq B$ for each tree decomposition $T = (B, F)$.*

The tree decomposition is only one way to characterize the treewidth of a graph. In the following, we present other characterizations of it and derive their upper and lower bounds for treewidth.

2.3.1 Forbidden Minors

A *forbidden minor characterization* is a method to characterize a set of graphs $\mathcal{G}$ by another set of graphs $\mathcal{F}$, called *forbidden minors*, such that a graph G is a member of $\mathcal{G}$ if and only if G does not have any of the graphs in $\mathcal{F}$ as minor. Robertson and Seymour showed that a graph class $\mathcal{G}$ can be characterized by a finite set of forbidden minors if $\mathcal{G}$ is closed under taking minors [RS04]. From the following Lemma 2.2 we can conclude that the partial k-trees, for some fixed $k \in \mathbb{N}$, are closed under taking minors, which is why they have a finite set of forbidden minors.

Lemma 2.2 ([Bod98, Lemma 16])**.** *If G' is a minor of G, then $tw(G') \leq tw(G)$.*

As an example, the forbidden minor sets of partial k-trees for $k = 1, 2$ only contain K_3 and the K_4 respectively. The forbidden minors of partial 3-trees are shown in Fig. 2.1.

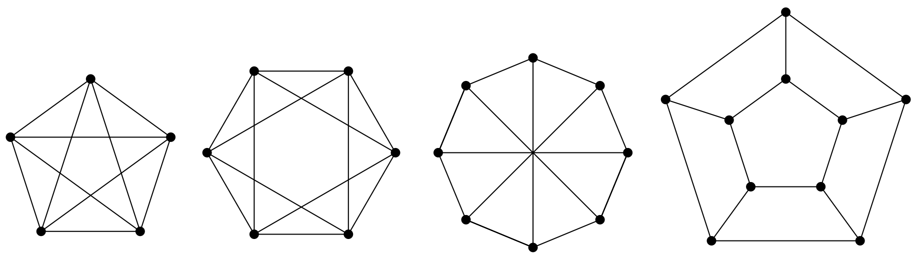

Figure 2.1: The four forbidden minors of partial 3-trees.

So, showing that a graph G has a forbidden minor of the partial k-trees as a minor proofs that the treewidth of G is at least $k + 1$. It is well known that the K_4 is a forbidden minor of the outerplanar graphs [Die12, Excercise 23], which is why they are closed under taking the minor and are a subclass of the partial 2-trees. A second crucial corollary from Lemma 2.2 is that the treewidth of an arbitrary minor of G is a lower bound to $\mathrm{tw}(G)$.

2.3.2 Brambles

In contrast to the other characterizations in this thesis, the *bramble characterization* is based on a maximization problem. The necessary terminologies from Boadlaender, Grigoriev, and Koster [BGK08] are introduced in the following.

Two vertex subsets $W_1, W_2 \subseteq V$ of a graph $G = (V, E)$ are *touching* each other, if they share a vertex or are adjacent, i.e., there is an edge $\{w_1, w_2\} \in E$ with $w_1 \in W_1$ and $w_2 \in W_2$. A set of mutual touching and connected vertex subsets is called a *bramble* of G. If a vertex subset $H \subseteq V$ intersects each element in a bramble $\mathcal{B}$, H is called a *hitting set* of $\mathcal{B}$. The size of a minimal hitting of a bramble is called its *order*. The maximal order over all brambles of a graph G is called the *bramble number* of G. The connection between the bramble number of a graph and its treewidth is given by Theorem 2.3.

Theorem 2.3 ([ST93]). *The treewidth of a graph is equal to its bramble number minus* 1.[1]

A corollary from Theorem 2.3 is that all brambles of a graph G imply a lower bound for the treewidth of G. This property is noted in the following.

Corollary 2.4 ([BGK08, Corollary 1.5]). *Let G be a graph with a bramble $\mathcal{B}$ of order k. Then $tw(G) \geq k - 1$ applies.*

2.3.3 Triangulation and chordal graphs

The following characterization of treewidth forms the basis for the upper bound heuristics we analyze in this thesis. For that, we introduce a special graph class, the *chordal* (or triangulated) graphs, on which the treewidth can easily be computed. According to Bodlaender, there are two equivalent ways to define the chordal graphs [BK10, Definition 3].

The first one is based on the idea of perfect elimination orderings. An *elimination ordering* of a general graph $G = (V, E)$ is a bijection $\pi : V \to \{1, \dots, |V|\}$. An elimination ordering π is called *perfect* if for every vertex $v \in V$ the set of *higher numbered neighbors* $N_G^\pi(v) \coloneqq \{w \in N_G(v) \mid \pi(w) > \pi(v)\}$ induce a clique in G. The chordal graphs are exactly those graphs that have a perfect elimination ordering. In other words, in every elimination step a vertex, whose whole neighborhood forms a clique, is eliminated. Such a vertex is called a *simplicial vertex.* A vertex v is called *almost simplicial* if at least $\deg(v) - 1$ neighbors of v induce a clique in G. The one vertex that is not included in the clique is called the *non-clique vertex* of v.

The second definition is that a graph is chordal if every cycle with at least four vertices has at least one chord, i.e., an edge between vertices that are not adjacent in the cycle [BK10, Definition 3]. One subclass of the chordal graphs are trees, since they do not have a circle of any size. It is well known that the treewidth of a chordal graph is equal to its clique number minus 1. Finding the maximal clique in an arbitrary graph is an $\mathcal{NP}$-hard problem, but it can be done on chordal graphs in linear time using Lemma 2.5.

Lemma 2.5. *Let $G = (V, E)$ be a chordal graph with perfect elimination ordering π. Then the treewidth of G is equal to* $\max_{v \in V} |N_G^\pi(v)|$.

Proof. The treewidth of a chordal graph $G = (V, E)$ is its clique number minus 1. So we have to proof that $\omega(G) - 1 = \max_{v \in V} |N_G^\pi(v)|$. Let C be a maximal clique in G and let $v \in C$ be the vertex in C with the smallest index according π. Then v has at least $\omega(G) - 1$ higher numbered neighbors, so $\omega(G) - 1 \leq \max_{v \in V} |N_G^\pi(v)|$ applies.

It remains to prove that $\omega(G) - 1 \geq \max_{v \in V} |N_G^\pi(v)|$ applies too. So, assume $\omega(G) - 1$ is strictly less than $\max_{v \in V} |N_G^\pi(v)|$. Then, there is a vertex $v \in V$ with $|N_G^\pi(v)| > \omega(G) - 1$, i.e., a vertex with at least $\omega(G)$ higher numbered neighbors. Since π is a perfect elimination ordering of G, $v \cup N_G^\pi(v)$ induce a clique in G with a size of $\omega(G) + 1$. Since this is a contradiction, $\omega(G) - 1 \geq \max_{v \in V} |N_G^\pi(v)|$ applies. $\square$

To characterize the treewidth of general graphs, we use the fact that all graphs can be „extended“ to a chordal graph. This is done by *triangulating* a graph. Formally a *triangulation* of a graph $G = (V, E)$ is a chordal graph $G^\Delta = (V, E^\Delta)$ with $E \subseteq E^\Delta$. It is additionally called *minimal* if it does not contain a triangulation of G with fewer edges.

Algorithm 2.1: TRIANGULATION

Input: Graph $G = (V, E)$, Elimination Ordering π
Output: Triangulation G^{Δ} of G

$G^{\Delta} \leftarrow G$
forall $i = 1$ *to* $|V|$ **do**
  $v \leftarrow \pi^{-1}(i)$
  $G^{\Delta} \leftarrow G^{\Delta} + clique(N^{\pi}_{G^{\Delta}}(v))$ // *fill-in* of v
return G^{Δ}

Using Algorithm 2.1 we can turn a graph G into a triangulation given an elimination ordering π.

Considering the transition from G^{Δ} to G in Algorithm 2.1, only edges are deleted. Therefore G is a minor of G^{Δ}, so $\mathrm{tw}(G) \leq \mathrm{tw}(G^{\Delta})$ applies (see Lemma 2.2). As a result, each triangulation of a graph G implies an upper bound to $\mathrm{tw}(G)$. Additionally, there is the following characteristic for partial k-trees:

Theorem 2.6 ([BK10, Theorem 6])**.** *Let $G = (V, E)$ be a graph and $k \in [1; |V|]$ an integer. Then the following are equivalent:*

i) G is a partial k-tree

ii) G has a triangulation G^{Δ} with $\omega(G^{\Delta}) \leq k + 1$

iii) There is an elimination ordering π of G, such that the graph FILLIN(G, π) *has a clique number of at most $k + 1$*

iv) There is an elimination ordering π of G, such that for every $v \in V$ the number of higher numbered neighbors in FILLIN(G, π) *is at most k*

A corollary from the equivalence „i) $\Leftrightarrow$ iv)“ is that every partial k-tree has at least one vertex with degree at most k.

2.4 Triangulation based heuristics

As determining the treewidth of an arbitrary graph is $\mathcal{NP}$-hard, heuristics are needed to, at least, approximate the treewidth of a graph. The ones we work with are based on the idea of triangulations introduced in Section 2.3.3.

Given a graph G these heuristics work on the space of all possible elimination orderings of G. Their target is to find a triangulation of G whose clique number is as small as possible, and is, therefore, a good upper bound of $\mathrm{tw}(G)$ (see Theorem 2.6). The heuristics are based on the same greedy approach shown in Algorithm 2.2, but use different priorities X to select the next vertex.

The operation described in line five of Algorithm 2.2 is called the *elimination* of the vertex v. It is the concatenation of the fill-in operation of v and the deletion of v. The graph obtained from G by the elimination of v is called the *elimination graph of G with respect to v*. If multiple vertices $A \subseteq V$ are eliminated one after another in G, the obtained graph is called the *elimination graph of G with respect to A*. The edges *involved* in an elimination are the edges that are either added during the fill-in of v or removed by the deletion of

[1]Note that the authors used the worlds „screen“ and „thickness“ instead of „bramble“ and „order“.

Algorithm 2.2: TW-HEURISTIC [IDEA]

Input: Graph $G = (V, E)$
Output: Elimination Ordering π

$G^{\Delta} \leftarrow G$
for $i = 1$ *to* $|V|$ **do**
  $v \leftarrow$ best vertex according to X in G^{Δ}
  $\pi(i) \leftarrow v$
  $G^{\Delta} \leftarrow (G^{\Delta} - v) + \text{clique}(N_{G^{\Delta}}(v))$ // *elimination* of v
return π

v. A (d, f)*-elimination* is an elimination of a vertex $v \in V$ with $\deg(v) = d$ and fill-in f. The parameter d and f are also referred to as the *degree* and *fill-in* of the elimination respectively. If the fill-in is not important in the context we omit the parameter f.

Besides the algorithm [BK10, Algorithm 3] Bodlaender and Koster also gave examples for possible priorities X. In this thesis we analyze the following two. If G_k^{Δ} is the graph after the k-th iteration, then the ...

- MINDEGREE selects one vertex with minimal degree in G_k^{Δ}.
- MINFILLIN selects one vertex with minimal fill-in in G_k^{Δ}.

To argue better about the heuristic we make some adaptions to Algorithm 2.2 that are explained in the following.

To make the heuristics deterministic we introduce a tiebreaker in case several vertices have the same priority. Therefore, an internal *vertex ordering* of the graph is given as a second input. Like an elimination ordering, vertex orderings are defined as a bijection $\rho : V \to \{1, \dots, |V|\}$, but it is used differently. Only if all vertices have the same priority, the vertex ordering equals the calculated elimination ordering.

The second adoption is that the heuristics also return the calculated upper bound for the treewidth. Yet the heuristics only return an elimination ordering, from which a triangulation has to be determined to get the approximated treewidth. This procedure can be shortened by simply returning the maximum over the degrees of the eliminated vertices (see Theorem 2.6 and Lemma 2.5). The final version of the heuristics is shown in Algorithm 2.3.

Algorithm 2.3: TW-HEURISTIC

Input: Graph $G = (V, E)$, Vertex Ordering ρ of G
Output: Elimination Ordering π, Treewidth tw

$G^{\Delta} \leftarrow G$
$\text{tw} \leftarrow 0$
for $i = 1$ *to* $|V|$ **do**
  $N \leftarrow$ set of vertices with optimal priority in G^{Δ} according to X
  $v \leftarrow$ vertex in N with lowest $\rho(v)$
  $\pi(i) \leftarrow v$
  $\text{tw} \leftarrow \max\{\deg(v), \text{tw}\}$
  $G^{\Delta} \leftarrow (G^{\Delta} - v) + \text{clique}(N_{G^{\Delta}}(v))$ // *elimination* of v
return tw, π

If G is a graph and ρ an arbitrary vertex ordering of G, we will denote the elimination ordering and the treewidth, calculated by a heuristic $X \in \{\text{MINDEGREE}, \text{MINFILLIN}\}$, with $\mathrm{X}(G,\rho)_\pi$ and $\mathrm{X}(G,\rho)_{\mathrm{tw}}$ respectively. If the vertex ordering we are referring to is clear from context, we omit the second argument from our notation, i.e., we write $\mathrm{X}(G)_\pi$ and $\mathrm{X}(G)_{\mathrm{tw}}$.

3. Comparison of the two Heuristics

A previous experiment showed how the two heuristics behave on real-world graphs [MSJ19a, Section 4]. One of the findings of this study was that the MinFillIn heuristic often finds better upper bounds than the MinDegree heuristic, while being a bit slower. While the running time of the MinFillIn heuristic was $\mathcal{O}(n^3 \log(n))$, the one of the MinDegree heuristic was only $\mathcal{O}(n^2 \log(n))$, where n is the number of vertices in the original graph [MSJ19b]. In this section, we show that neither of the two heuristics is always better than the other one. For that, we construct a family of graphs on which the MinDegree can perform better than the MinFillIn heuristic, and the other way round. We also show that the MinDegree can be arbitrarily far away from the actual treewidth of a graph.

3.1 MinFillIn better than MinDegree

In Section 5.2 we show that that the MinDegree heuristic does not only work on all $2\times n$-grids but also on all partial 2-trees. In this section, we show that such a statement can not be made upon graphs with higher treewidth. For this, we define the graph family $G_{k,l}$ with $k, l \in \mathbb{N}$. The graph $G_{k,l}$, for some fixed k and l, consists out of a K_k and l almost simplicial vertices that are connected to all vertices in the K_k and share the same non-clique-vertex v. The structure of these graphs is shown in Figure 3.1.

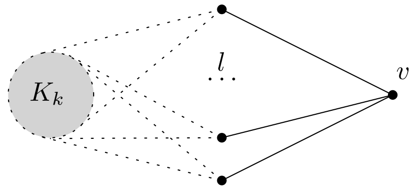

Figure 3.1: General structure of the graph $G_{k,l}$.

We use these graphs, to build a graph family on which the MinDegree heuristic can work arbitrarily bad, and on which the MinFillIn heuristic returns the actual treewidth. We thereby give a graph family on which the MinFillIn heuristic returns a better result than the MinDegree heuristic. As a part of this proof, we show the treewidth of some of those graphs with the help of the MinFillIn heuristic in Lemma 3.1.

Lemma 3.1. *Let $k, l \leq 2$ with $k \leq l(l-1)/2$, then $tw(G_{k,l}) = k+1$.*

Proof. For this proof, we show the lower bound by finding a K_{k+2} minor and an upper bound by the MinFillIn heuristic.

If we contract an arbitrary edge that is incident to the vertex v, we get a graph with a K_{k+2} subgraph consisting out of K_k, the vertex from the edge contraction and an untouched almost simplicial vertex. So $G_{k,l}$ has a K_{k+2} minor and therefore a treewidth of at least $k+1$.

For the upper bound we use the MinFillIn heuristic. The fill-in of all almost simplicial vertices is k, while all other fill-ins are exactly $l(l-1)/2$, i.e., the number of edges that are necessary to connect the independent set of all almost simplicial vertices. As k is at most $l(l-1)/2$ by assumption, the MinFillIn heuristic can eliminate one of the almost simplicial vertices. By this elimination, all remaining almost simplicial vertices are now simplicial, which is why all of them but one are eliminated in the next elimination steps. The result of those eliminations is a K_{k+2}, so the MinFillIn heuristic returns an upper bound of $k+1$. The steps of the MinFillIn heuristic are illustrated in the Figure 3.2.

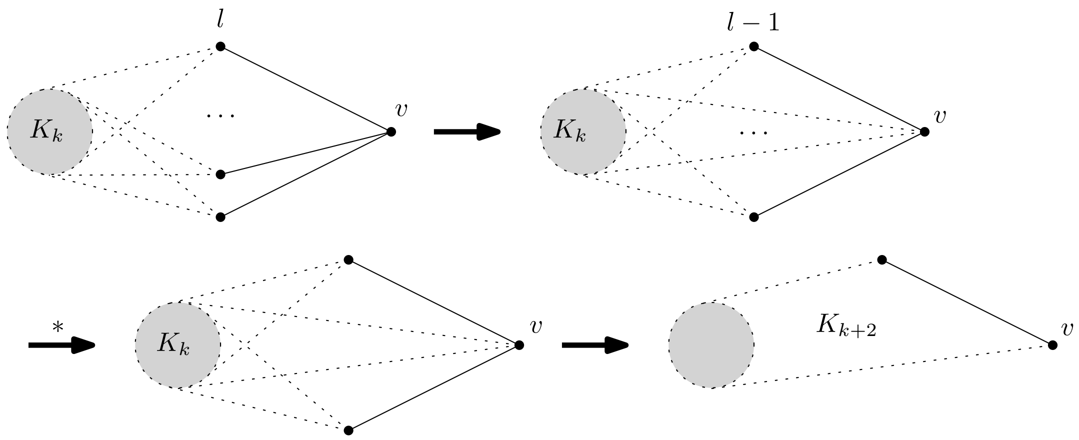

Figure 3.2: Elimination steps of the MinFillIn heuristic on the graph $G_{k,l}$ with $k \leq l(l-1)/2$.

□

In the following Theorem 3.2 we give example graphs on which the MinDegree heuristic does not return the correct result, and on which the difference between the result and actual values increases with increasing treewidth.

Theorem 3.2. *For every $k \geq 3$ there is a partial k-tree G with vertex ordering ρ, such that* MinDegree$(G,\rho)_{tw} > k$*. For every $k \geq 4$ there is even a graph, such that this statement holds for every vertex ordering ρ of G, and* MinDegree$(G)_{tw} - k \in \mathcal{O}(k)$*.*

Proof. For $k = 3$ the example graph is $G_{2,3}$, whose treewidth is exactly 3 by Lemma 3.1. In this graph, both vertices in the clique have a degree of 4, and all other vertices have a degree of 3. If v has the lowest number in ρ, then v is eliminated first, which results in a K_5. In this case, the MinDegree heuristic returns an upper bound of four.

For $k \geq 4$ consider the graph $G_{k-1,k-1}$. By Lemma 3.1 the treewidth of $G_{k-1,k-1}$ is exactly k, because of

$$k-1 \leq 1 \cdot (k-1) \overset{k \leq 4}{\leq} \frac{k-2}{2}(k-1).$$

The minimum degree of $G_{k-1,k-1}$ is $k-1$ and is only attained by v. After the elimination of v the remaining graph is complete with $2(k-1)$ vertices, which is why MinDegree heuristic returns an upper bound of $2k-3$. The difference between k and $2k-3$ increases linear in k. □

Corollary 3.3. *For every $d \geq 1$ there is a graph G, such that the* MinDegree *heuristic returns a result that is d greater than the result of the* MinFillIn *heuristic on G.*

Proof. Let $d \geq 1$ and $k := d+3$. Now consider the graph $G_{k-1,k-1}$. In the proof of Theorem 3.2 we show that the MinDegree heuristic returns a result of $2k-3 = 2d+3$. In the proof of Lemma 3.1 we show that the MinFillIn heuristic returns the correct treewidth of $k = d+3$. The difference between these two results is exactly d, which proves this corollary. □

3.2 MinDegree better than MinFillIn

In this section, we first find a partial 2-tree such that the MinFillIn heuristic can return a higher value than 2. As the MinDegree heuristic always return 2 for every partial 2-tree (Section 5.2), this graph disproves the claim that the MinFillIn heuristic is always better than the MinDegree heuristic. After that, we show a technique to create graphs, and thereby a family of graphs, of arbitrary higher treewidth with the same properties.

Lemma 3.4. *It exists a graph G with a vertex ordering ρ of G, such that* $\text{MinFillIn}(G,g)_{tw}$ *is strictly greater than* $\text{MinDegree}(G,\rho)_{tw}$.

Proof. Let G be the graph shown in Figure 3.3 and let v be the first vertex in the vertex ordering ρ. Then v is eliminated first by the MinFillIn heuristic, which is why it returns a value of at least 3. As G is a partial 2-tree, the MinDegree heuristic returns the correct treewidth of 2 (Theorem 5.8).

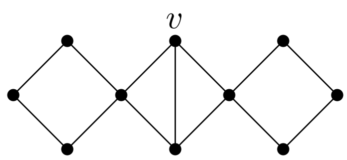

Figure 3.3: Example of a partial 2-tree for that the MinFillIn heuristic can return a value of 3.

□

Based on the graph presented in Lemma 3.4, we build a whole graph family such that for every graph in this family the MinFillIn heuristic can return a higher result than the MinDegree heuristic. This graph family is successively created by adding a vertex that is adjacent to all other vertices. This operation is inspired by the idea of David Eppstein [Epp11]. In the following we show the treewidth of the extended graph in Lemma 3.5 and the behavior of both heuristics on this graph in Lemmas 3.6 and 3.7.

Lemma 3.5. *Let $G = (V,E)$ be a graph with treewith k, and let G^+ be the graph G with an additional vertex x that is connected to all vertices in V. Then G^+ has a treewidth of $k+1$.*

Proof. In this proof, we use brambles to show the lower bound and tree decompositions for the upper bound.

Since G has treewidth k it has a bramble $\mathcal{B}$ with order $k+1$ (Theorem 2.3). We use this bramble of G to construct a bramble of G^+ with an order of $k+2$. This bramble $\mathcal{B}^+$ consists out of the vertex subsets in $\mathcal{B}$ and the set $\{x\}$. To prove that $\mathcal{B}^+$ is indeed a bramble of G^+, we show that every two sets $X, Y \in \mathcal{B}^+$ are touching each other. If X and Y are part of $\mathcal{B}$ they touch each other because G is a subgraph of G^+ and $\mathcal{B}$ is a bramble of G. If one of the sets is $\{x\}$, w.l.o.g. X, and $Y \in \mathcal{B}$, then X is touching Y in G^+ because v is adjacent to every vertex in G and therefore also to a vertex in Y. It remains to show that $\mathcal{B}^+$ has a minimal hitting set with $k+2$ vertices. We know that $\mathcal{B}$ already has a minimal hitting set $H \subseteq V$ with $k+1$ vertices, which does not include x. Therefore $H \cup \{x\}$ is a minimal hitting set for G^+ with $k+2$ vertices. As a result, the bramble number of G is $k+2$, so G^+ has a treewidth of at least $k+1$.

To show that the treewidth of G^+ is at most $k+1$, we construct a tree decomposition for G^+ with width $k+1$ using a minimum tree decomposition of G. Since G has treewidth k, it has a minimum tree decomposition $T = (B, F)$ with width k and therefore only bags with a size of at most $k+1$. We show that $T^+ := (B^+, F)$, where B^+ is the set of bags B in which x is added to each bag, is a tree decomposition of G^+. For that, we show that adding x to each bag does not change the properties of a tree decomposition. Since x is included in all bags, the union of all bags in B^+ covers all vertices in V as well as x. Additionally, there is at last one bag in B^+ for each neighbor v of x, such that v and x are included in that bag, because the union of all bags in B covers all vertices in V. Finally, the graph induced by all bags that include x in T^+, i.e., the whole graph T^+, is a tree, because T is a tree by definition and T is isomorphic to T^+. $\square$

Using Lemma 3.5 we can successively extend the graph given in the proof of Lemma 3.4 to a graph family with graphs of arbitrary large treewidth as shown in Figure 3.4.

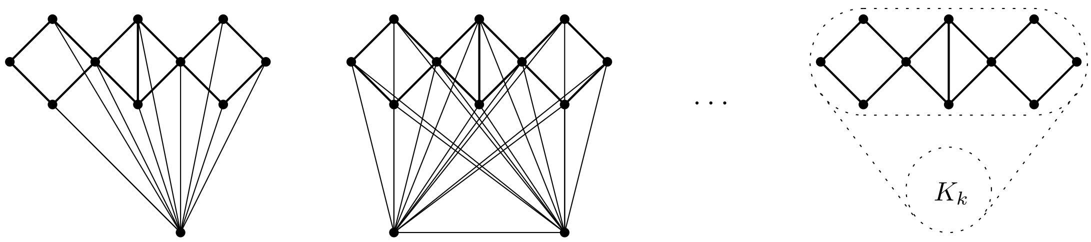

Figure 3.4: Extending the graph from Lemma 3.4 to a family of graphs. In the graph on the right side all vertices in the clique K_k are connected to all vertices in the graph above.

Now, we show that G and G^+ from the previous lemma have similar properties with regards to the MINFILLIN heuristic, in the following sense:

Lemma 3.6. *Let $G = (V, E)$ and let G^+ be the graph G with an additional vertex x that is connected to all other vertices. If there is a vertex ordering ρ of G such that* $\text{MINFILLIN}(G, \rho)_{tw} = k$ *then there is also a vertex ordering ρ^+ of G^+ such that* $\text{MINFILLIN}(G^+, \rho^+)_{tw} = k+1$.

Proof. Let $\rho^+(v) := \rho(v)$ for all $v \in V$ and define $\rho^+(x) := |V| + 1$. We show that the elimination sequences are identical for all vertices in V, and that x is eliminated last in $\text{MINFILLIN}(G^+, \rho^+)$. Note that all vertices in V have the same fill-in G^+ as they have in G, because v is adjacent to all vertices in the G-subgraph of G^+. As a basis for our further argumentation, we show that a vertex that is connected to all other vertices always has the highest fill-in in a graph.

Let v be an arbitrary vertex in V and let x be a vertex that is connected to all vertices in V. If we eliminate v no edge is added that is incident to x, because x already has the maximum number of incident edges. Therefore the set of all fill-in edges F is added to the subgraph $G^+[N_{G^+}(v) \setminus \{x\}] = G[N_G(v)]$. Since $N_G(v) \subseteq V = N_{G^+}(x)$, F is a subset of all fill-in edges that are added during the elimination of x in G^+. As a result, the fill-in of x in G^+ is at least as high as the fill-in of an arbitrary vertex, and thus a maximum fill-in.

Now consider the first elimination step in $\textsc{MinFillIn}(G^+, \rho^+)$. If we assume, that G^+ has at least two vertices, we know that there is another vertex besides x whose fill-in is at most the fill-in of x. Since x has the highest number in the vertex ordering ρ^+, x is not chosen in this step. Additionally, we know that the fill-in of all vertices in V is the same as in G. Therefore $\textsc{MinFillIn}(G^+, \rho^+)$ eliminates the same vertex as $\textsc{MinFillIn}(G, \rho)$ does. The elimination graph of this step has again a vertex that is connected to all other vertices, namely x. So we can use the same argumentation as in the first step for all further elimination steps. As a result, $\textsc{MinFillIn}(G^+, \rho^+)$ eliminates the vertices in V in the same order as $\textsc{MinFillIn}(G, \rho)$ does, and eliminates x last. The only difference is, that the degree of all vertices in V is increased by 1 due to the additional edge to x. Therefore the result of $\textsc{MinFillIn}(G^+, \rho^+)$ is 1 higher than the result of $\textsc{MinFillIn}(G, \rho)$. $\square$

Using Lemma 3.6, we show that the graph family given in Figure 3.4 has the property that the MinFillIn heuristic does not always work optimally on it. Therefore, there is a graph for each $k \geq 3$ such that the MinFillIn heuristic can return a higher value than the actual treewidth of this graph. With the help of the following Lemma 3.7 we can additionally state that the graph family in Figure 3.4 contains examples on which the MinDegree heuristic performs better than the MinFillIn heuristic.

Lemma 3.7. *Let $G = (V, E)$ and let G^+ be the graph G with an additional vertex x that is connected to all other vertices. If there is a vertex ordering ρ of G such that $\textsc{MinDegree}(G, \rho)_{tw} = k$ then there is also a vertex ordering ρ^+ of G^+ such that $\textsc{MinDegree}(G^+, \rho^+)_{tw} = k + 1$.*

Proof. We prove this lemma the same way as Lemma 3.6. So let $\rho^+(v) := \rho(v)$ for all $v \in V$ and $\rho^+(x) := |V| + 1$. Again, we show that a vertex that is connected to all other vertices has the lowest priority, i.e., the highest degree. Then, we use the same argumentation from the proof of Lemma 3.6 to derive this lemma. So assume that there is a vertex v with a higher degree than $\deg(x)$. Since x is connected to all other vertices, the vertex v has to be connected to all other vertices and itself, which is a contradiction to the assumption that G, and therefore G^+, is simple. $\square$

This technique can be used to extend a graph to a whole family of graphs, such that each one of them adopts the graph's properties with respect to the behavior of both heuristics. As a result, as soon as one finds a single graph G on which for example two possible results of the MinDegree and the MinFillIn heuristic are more than 1 apart, one can give a whole family of graphs with that property, covering all treewidths above $\mathrm{tw}(G)$.

4. The MinFillIn Heuristic on Small Grids

One main focus of this thesis is the study of the heuristics behavior on grids. An $m{\times}n$-*grid* is the graph with vertices $V = \{(i,j) \mid i = 1 \ldots m, j = 1 \ldots n\}$, where every two vertices $(i_1, j_1), (i_2, j_2) \in V$ are adjacent if $||(i_1, j_1) - (i_2, j_2)||_1 = 1$. The treewidth of an $m{\times}n$-grid is proven by Chlebikova [Chl92, Theorem 16] and Bodlaender [Bod98, Corollary 89] to be $\min\{n, m\}$. A planar drawing of a general grid is given in Figure 4.1. All vertices that lie on the outer face of this drawing form the grid's *border*. The values of n and m are called the *side lengths* of the grid.

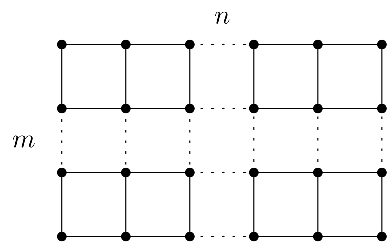

Figure 4.1: The general structure of a $m{\times}n$-grid.

In the following three sections we show that the MinFillIn heuristic returns the correct treewidth for every $m{\times}n$-grid with $n \in \mathbb{N}$ and $m = 1, 2, 3$ respectively. We even show that this applies to certain superclasses of $1{\times}n$-grids and $2{\times}n$-grids respectively. Before we deal with these graph classes, we show two simple facts that are needed for many proofs of this thesis, but are omitted for reasons of clarity in all following sections.

Lemma 4.1. *The connected graphs are closed under eliminations.*

Proof. Let $G = (V, E)$ be a connected graph, $v \in V$ a vertex and G' their elimination graph. Further, let $u, w \in V \setminus \{v\}$ be two connected vertices in G and $S = (u = s_1, \ldots, s_p = w)$ a path between them. If v is not included in S the same path exists in G' as only edges incident to v are removed during the elimination. If, otherwise, v is part of S at index $i \in \{2, \ldots, p-1\}$ the path can be modified to exist in G' in the following way. At first, $s_i = v$ have to be removed from S, as v does not exist in G' anymore. It remains a path $S = (u = s_1, s_2 \ldots, s_{i-1}, s_{i+1}, \ldots, s_{p-1}, s_p = w)$. Because s_{i-1} and s_{i+1} were neighbors of v in G, they are adjacent in G' after the elimination of v (in particular after its fill-in). Therefore, S' is already a valid path between u and w in G'. Because u and w are chosen arbitrarily from the set of vertices of G', the graph G' is connected. $\square$

Lemma 4.2. *The connected graphs are closed under edge contractions.*

Proof. Let $G = (V, E)$ be a connected graph, and let $G' = (V', E')$ be the graph G after contracting an edge $\{x, y\} \in E$ to a vertex $z \in V'$. To prove that G' is connected, we show that every pair of vertices $u, v \in V$ is connected in G'. If there is a path between u and v in G that neither includes x nor y, then this path exists in G' too. Otherwise, let P be a path between u and v without cycles. We know that z is connected in G' to all neighbors of x and y in G. Therefore P', the path that remains after replacing all occurrences of $(\dots, x, y, \dots)$, $(\dots, x, \dots)$ and $(\dots, y, \dots)$ by $(\dots, z, \dots)$ in P (in that order), is a path in G'. □

4.1 Chordal Graphs and Trees

To prove that the MinFillIn heuristic returns the correct treewidth for every $1 \times n$-grid, we show that this even applies to the superclass of chordal graphs. An important part of this proof is to show that the chordal graphs are closed under elimination (Corollary 4.4). For this, we show that an elimination does not increase the size of an induced cycle in the following sense.

Lemma 4.3. *Let $G = (V, E)$ be a graph, $v \in V$ a vertex and $G' = (V', E')$ their elimination graph. Further, let $C' \subseteq V'$ be a vertex subset with $|C'| \geq 4$ that induces a chordless cycle in G'. Then there is a set of vertices $C \subseteq V$ in G such that $C' \subseteq C$ and $G[C]$ is a chordless cycle.*

Proof. Let G, v, G' and C' be chosen as in the lemma. Now there are three cases to be considered:

- If $|C' \cap N_G(v)| \leq 1$, then C' induces a cycle in G too, because no edge in $G[C']$ could be involved in the elimination of v.
- If $C' \cap N_G(v) = \{u, w\}$ and $\{u, w\} \in E$, then C' induces a cycle in G too. The only edge that is involved in the elimination of v is the edge $\{u, w\}$, as it could be added if it did not already exist in G. But because of $\{u, w\} \in E$, this is not the case.
- If $C' \cap N_G(v) = \{u, w\}$ and $\{u, w\} \notin E$, then C' only induces a path in G, because all edge but $\{u, w\}$ are not involved in the elimination of v. From $u, w \in N_G(v)$ we know, that G contains the edges $\{u, v\}$ and $\{v, w\}$. As $\{u, w\} \notin E$ the set $C' \cup \{v\}$ induces a cycle in G that is even larger than the cycle $G'[C']$.

As a result, there is a set $C \subseteq V$ fulfilling the conditions in the lemma for each case. Note that this case distinction is complete, because if $|C' \cap N_G(v)|$ was higher than 2, there was at least one chord in $G'[C']$. So C' would not induce a cycle in G'. □

Using Lemma 4.3 we now show that chordal graphs are closed under eliminations.

Corollary 4.4. *The chordal graphs are closed under elimination.*

Proof. Let $G = (V, E)$ be a chordal graph and $v \in V$ an arbitrary vertex in G. Because G is chordal, the maximum size of an induced cycle in G is 3. From Lemma 4.3 we know, that the elimination graph G' of G with respect to v does not have an induced cycle of length 4 either. Therefore G' is also chordal. □

It remains to prove that any possible elimination in the MinFillIn heuristic does not increase the largest clique in a graph. This is the key part of the following proof.

Theorem 4.5. *The* MinFillIn *heuristic returns the correct treewidth for all chordal graphs.*

Proof. Let $G = (V, E)$ be a chordal graph and let $v \in V$ be a vertex with minimum fill-in in G. We show that the elimination graph of G and v is a chordal graph that does not have a larger maximum clique than G. This implies that in each elimination step of the MinFillIn heuristic only a vertex with a smaller degree than the treewidth of G is eliminated.

It can be seen, that the minimum fill-in of a chordal graph is 0, since there is always a simplicial vertex by definition. Therefore v has to be a simplicial vertex. As a simplicial vertex with degree k implies a clique with size $k+1$, and such a clique implies a treewidth of at most k, the eliminated vertex v has a degree of at most $\mathrm{tw}(G)$. We additionally know that G' is a proper subgraph of G, since edges and vertices are only removed and not added during the elimination of v. The graph G' is also chordal by Corollary 4.4.

As a result, only vertices with a degree of at most $\mathrm{tw}(G)$ are eliminated during the MinFillIn heuristic. Therefore, $\text{MinFillIn}(G)_{\mathrm{tw}}$ is at most $\mathrm{tw}(G)$, and, because it is an upper-bound heuristic, exactly $\mathrm{tw}(G)$. ☐

A direct corollary from Theorem 4.5 is that the MinFillIn heuristic returns 1 for every tree, since trees are a subclass of chordal graphs. This additionally applies to all $1\times n$-grids, since they are just paths, which is a subclass of trees.

4.2 2xn-Grids

In this section, we show that the MinFillIn heuristic works optimally on all $2\times n$-grids. In Lemma 3.4 we show that there is a partial 2-tree, on which the MinFillIn heuristic does not return 2. Therefore we need to narrow down the graph class for which this is true. The superclass of $2\times n$-grids we consider are the outerplanar 2-connected graphs with maximum degree 3, abbreviated with $\mathcal{G}_2$. To achieve this, we show that the minimum fill-in of a graph in $\mathcal{G}_2$ is at most 1 in Lemma 4.6, and that $\mathcal{G}_2$ is closed under all possible eliminations with this fill-in in Lemma 4.7.

Lemma 4.6. *Let G be an outerplanar 2-connected graph with maximum degree 3. Then the minimum fill-in of G is 1.*

Proof. As a member of the outerplanar graphs, G also has a treewidth of at most 2. Since every graph with treewidth k has at least one vertex with degree at most k, every outerplanar graph has a vertex with degree at most 2. Because vertices with a degree of at most 2 have a fill-in of at most 1, the minimum fill-in of G is at most 1. ☐

Lemma 4.7. *Let $G = (V, E)$ be an outerplanar 2-connected graph with maximum degree 3 (called $\mathcal{G}_2$), and let $v \in V$ be a vertex with minimum fill-in. Then G', the elimination graph of G and v, is either a tree or in $\mathcal{G}_2$.*

Proof. To rule out a simple base case, note that if G is a chordless cycle, every possible elimination graph of G is also a chordless cycle. So the statement holds in this case. In the following, we make a case distinction over the minimum fill-in of G.

If the minimum fill-in of G is 0, a simplicial vertex is eliminated in the next step. Let $v \in V$ be this vertex, whose degree can only be 2 or 3 by definition of $\mathcal{G}_2$. If the degree of v was 3 there would be a K_4-subgraph in G, which is the forbidden minor for treewidth 2, i.e., the treewidth of G. If $\deg(v) = 2$ then G is either K_3, i.e., a chordless cycle, or v is adjacent to two vertices u and w that are connected by a chord. As shown in Figure 4.2, this chord closes the cycle after the elimination of v. Note that elimination of v does neither change any other vertices than u,v and w, nor the degree of u and w. As a result the elimination graph is a member of $\mathcal{G}_2$.

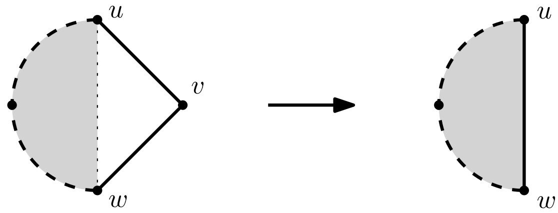

Figure 4.2: An outerplanar 2-connected graph G with maximum degree of 3 (left) and the elimination graph of G with respect to a vertex with fill-in 1 and degree 2 (right). The gray areas are placeholders for an arbitrary number of chords.

Now, let the fill-in of v be 1. Again, only the cases $\deg(v) = 2$ and $\deg(v) = 3$ have to be considered. The case $\deg(v) = 2$ is already covered by Figure 4.2, if we ignore the dotted chord before the elimination. In the following, we show by contradiction that case $\deg(v) = 3$ can not occur. So assume v has a degree of 3 and a minimum fill-in of 1 in G. The only way v can have a degree of 3, is that it is incident to two edges on the cycle of the outerplanar graph G and one chord. Let $u, w \in V$ be the neighbors of v on the cycle and $x \in V$ the vertex on the other side of the chord. Since there is a chord in G, there are at least four vertices, which is why u and w are not adjacent. But because v has a fill-in of 1, these vertices have to be adjacent to x. Therefore, we already know all neighbors of x in G. As a result, the graph illustrated in Figure 4.3 is the only possible graph for this situation.

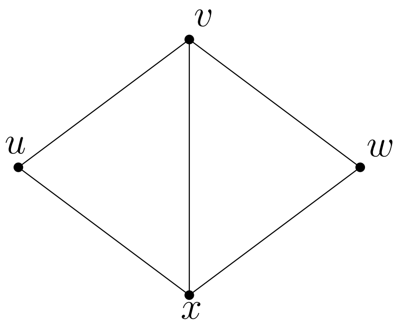

Figure 4.3: The only graph in which v has a degree of 3 and a fill-in of 1.

As a result, both u and w are vertices with fill-in 0, which is a contradiction to v being a vertex with minimum fill-in. □

Corollary 4.8. *The* MinFillIn *heuristic returns the correct treewidth for all* $2{\times}n$*-grids.*

Proof. From Theorem 4.5 we know that this statement applies for all $1{\times}n$-grids. So it only remains to prove that the MinFillIn heuristic returns a value of at most 2 for every $2{\times}n$-grid with $n \geq 2$. For this, we use the previous lemmas. These state that in a graph of $\mathcal{G}_2$ only vertices with a degree of at most 2 are eliminated by the MinFillIn heuristic, and that $\mathcal{G}_2$ is closed under these eliminations. As members of $\mathcal{G}_2$ this applies to all $2{\times}n$-grid with $n \geq 2$ and their elimination graphs under the MinFillIn heuristic. □

In the following, we show that there is at least a vertex ordering for every partial 2-tree, such that the MinFillIn returns the correct treewidth.

Lemma 4.9. *For every partial* 2*-tree* G *there is a vertex ordering* ρ *such that the* MinFillIn *heuristic returns a value that is at most* 2*.*

Proof. In this proof we use the MinFillIn heuristic with a minimum-degree-tiebreaker instead of a vertex ordering for G, i.e., if there are several vertices with the same (minimum) fill in, the one with the smallest degree is chosen. We show that this alternated heuristic only eliminates vertices with a degree of at most 2. The resulting elimination sequence can then be used as the ρ mentioned in the lemma.

Since G is a partial 2-tree, there is at least one vertex with degree at most 2. Therefore, the only chance that a vertex with degree 3 or higher is chosen by this alternated heuristic is, that this vertex has a fill-in of 0 and all other have a higher fill-in. This is not possible because such a vertex implies a K_4 minor in G, i.e., a forbidden minor for partial 2-trees. Therefore, one of the vertices with degree of at most 2 is eliminated in the first step. Since all eliminations with degree at most 2 can be seen as an edge contraction, the remaining graph G' is a minor of G and therefore again a partial 2-tree. We can show inductively with the same argumentation that this alternated heuristic only eliminates vertices with degree of at most 2 in its course on G. □

4.3 3xn-Grids

In contrast to the previous two sections of this chapter, we only focus on the grids themselves in this section. For this, we give a step-by-step argumentation of why the MinFillIn heuristic only eliminates vertices with degree at most 3 on $3{\times}n$-grids in the following Lemma 4.10.

Lemma 4.10. *The* MinFillIn *heuristic returns the correct treewidth for all* $3{\times}n$*-grids.*

Proof. For $n \leq 2$ the proof is already done by Corollary 4.8. For $n \geq 3$, we show that the MinFillIn heuristic only eliminates vertices on the short side of the grid (Figure 4.4), until only two rows of vertices remain.

At the beginning, all corners have a fill-in of 1, every border vertex has a fill-in of 3 and every inner vertex has a fill-in of 6. To show that only vertices on the short side are eliminated, we show that every elimination has a fill-in that is strictly less than 3, because otherwise arbitrary border vertices could be eliminated. The elimination steps of the MinFillIn heuristic are shown in Figure 4.4. At first, both corner vertices on the short side are eliminated, which results in a $3{\times}(n-1)$-grid with a vertex, that is connected to all three vertices in the next row, called a *peak*. After the elimination of this peak, there are two different kinds of vertices with a minimum fill-in, that are marked with „1“ and „2“. No matter which of those vertices is chosen, the next elimination results again in a grid with a peak, but this time with one row less, as shown in Figure 4.4.

These, elimination steps can happen independently on both 3-sides of the grid, until both sides are too close to each other. In this case, two rows of vertices remain. The elimination steps for this graph are shown in Figure 4.5. The left graph consists of two 3-cliques whose vertices have exactly one edge that is incident to the other clique. Therefore the eliminations of all vertices would result in the same graph shown in the middle. This graph is a 4-cycle with a vertex that is connected to all vertices in the cycle. Since the single vertex has a fill-in of 2 and the vertices on the cycle only have a fill-in of 1, one of them is eliminated in the next step of the MinFillIn heuristic. The elimination graph of this

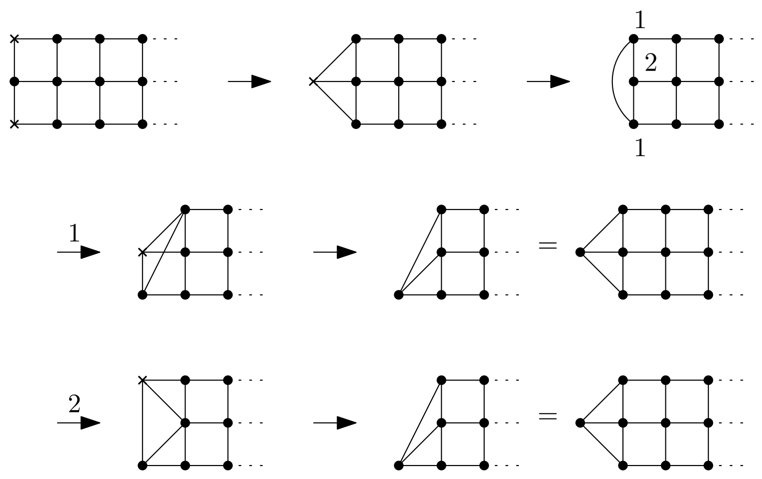

Figure 4.4: Elimination steps of the MinFillIn heuristic on a $3 \times n$-grid. The vertices marked with a cross are eliminated in the next step. If multiple vertices can be eliminated in the next step, they are marked with different numbers.

elimination is a complete graph with four vertices, which is why no vertices with a degree higher than 3 can be eliminated in the remaining eliminations. Since no vertices with a degree of 4 or higher were eliminated, the MinFillIn heuristic returns a value of 3, i.e., the correct treewidth of a $3 \times n$-grid.

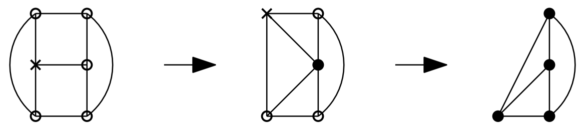

Figure 4.5: Last elimination steps of the MinFillIn heuristic on a $3 \times n$-grid. The vertices marked with a cross are eliminated in the next step. The elimination of a vertex that is marked with a cycle would result in the same graph as the elimination of the „cross“-vertex.

□

5. The MinDegree Heuristic on Grids

In the course of this chapter we often prove, that, for a graph G from a graph family $\mathcal{G}$, MinDegree$(G)_{\text{tw}}$ is at most k for some $k \in \mathbb{N}$. A sufficient condition for that is that $\mathcal{G}$ is closed under all minimum-degree eliminations with a degree at most k. Under this condition, only vertices with a degree of at most k are eliminated by the MinDegree heuristic, which is why MinDegree$(G)_{\text{tw}}$ is at most k.

5.1 Chordal Graphs

As with the other heuristic, we also show for the MinDegree heuristic that it performs optimally on all chordal graphs, and therefore also on all $1\times n$-grids. In Corollary 4.4 of the previous chapter, we show that the class of chordal graphs is closed under elimination. It remains to prove that all possible eliminations with respect to the MinDegree heuristic do not increase the clique number $\omega(G)$ of a chordal graph G. This is shown in Lemma 5.2 with the help of the following Lemma 5.1.

Lemma 5.1. *Let $G = (V, E)$ be a chordal graph, then $\delta(G) < \omega(G)$ applies.*

Proof. Assume $\delta(G) \geq \omega(G)$. Since G is a chordal graph, it has a perfect elimination ordering and therefore has a simplicial vertex $x \in V$. Since $\delta(G)$ is the minimum degree of G, the degree of x is at least $\delta(G)$. Because x is simplicial, it is part of a clique with size $\delta(G) + 1$. Because of the assumption, this is a clique with a larger size than the clique number of G, which is a contradiction. □

Lemma 5.2. *If $G = (V, E)$ is a chordal graph and $v \in V$ a vertex with minimum degree in G. Then the elimination of v does not create a clique with size higher than* $\deg(v) + 1$.

Proof. Let G' be the elimination graph of G with respect to v. Now assume the elimination of v creates a clique K_l with $l > \deg(v) + 1$ in G' that does not exist in G. Therefore, the elimination creates at least one edge $\{u_1, u_2\}$ that is not in G. Moreover, because the neighbors of v just form a clique with size $\deg(v)$ after the elimination of v, there is at least one vertex $w \in K_l \setminus N_G(v)$. Since K_l is a clique and the elimination can not create edges that are incident to w, the vertex w has to be adjacent to all neighbors of v, including u_1 and u_2, already in G. Now, it is shown that under the assumption there is a chordless cycle (v, u_1, w, u_2, v) with size 4 in G, because $\{u_1, u_2\}$ is not an edge in G and w is not in the neighborhood of v. The existence of such a cycle is a contradiction to G being a chordal graph. □

Using the previous Lemma 5.2, we now show that MinDegree heuristic performs optimally on all chordal graphs in Theorem 5.3. As all $1 \times n$-grids are chordal, this also applies to them.

Theorem 5.3. *The* MinDegree *heuristic returns the actual treewidth of a chordal graph.*

Proof. Let $G = (V, E)$ be a chordal graph and let $v \in V$ be a vertex with minimum degree. We know from Lemma 5.1 that the minimum degree of G is at most its clique number minus one, which is exactly the treewidth of G, i.e., $\delta(G) \leq \omega(G) - 1 = \text{tw}(G)$ applies. For the proof of this lemma, we show that the elimination graph of G with respect to v has again a minimum degree of at most $\text{tw}(G)$. This allows us to prove inductively that the MinDegree heuristic only eliminates vertices with degree of at most $\text{tw}(G)$ in G.

Let G' be the elimination graph of G with respect to v. We know that the degree of v is at most the treewidth of G. Therefore, the elimination of v does not create a clique with more than $\text{tw}(G) + 1$ vertices (Lemma 5.2). As such a clique does not exist in G either, every maximal clique in G' has a size of at most $\text{tw}(G) + 1$. Because G' is chordal by Corollary 4.4, the treewidth of G' is at most $\text{tw}(G)$. As a result, there is at least one vertex with degree of at most $\text{tw}(G)$ in G'. □

5.2 Partial 2-Trees

In this section, we show that the MinDegree heuristic is not optimal for all partial 2-trees, and therefore also for all $2 \times n$-grid. This statement is shown in Theorem 5.8. For this, we introduce another characterization of partial 2-trees in Definition 5.4.

Definition 5.4. A *DSP-graph* is a graph that can be constructed by the following rules:

- the K_2 is a DSP-graph
- if $G = (V, E)$ is a DSP-graph, then the following operations result in a DSP-graph
  - **D**: adding a leaf (also called *dangling vertex*) to a DSP-graph result in a DSP-graph
  - **S**: replacing an edge $\{a, b\} \in E$ by two edges $\{a, c\}$ and $\{c, b\}$ with $c \notin V$. This procedure is also called a *subdivision* of the edge $\{a, b\}$
  - **P**: adding two edges $\{a, c\}$ and $\{c, b\}$ with $c \notin V$ for an existing edge $\{a, b\} \in E$

Note that the defined operations D, S and P are the inverses of all possible eliminations the MinDegree heuristic may perform if the input graph has a minimum degree of 2 or lower.

Lemma 5.5. *A graph is a* 2*-tree if and only if it is a DSP-graph.*

Proof. Let G be a graph with $\text{tw}(G) \leq 2$. Then there is a minimum triangulation G^{Δ} of G with maximum clique size 3 (see Theorem 2.6). Furthermore, there is a perfect elimination ordering π of G^{Δ}, such that $2 = \text{tw}(G) = \max_{v \in V} |N_{\pi}(v)|$ (Lemma 2.5). Now, if we eliminate all vertices step by step from G according to π, we only use the inverse operations of D,S or P per definition. Therefore there is an operation sequence that constructs G from K_2.

We prove the other implication by induction over the number of vertices. In the base case the K_2 has a treewidth of 1. Now assume that all DSP-graphs with n vertices have a treewidth of 2 or lower. Let $G = (V, E)$ be a DSP-graph with $|V| = n+1$ vertices for $n \geq 2$. Per definition there is a sequence $S = (s_1, \ldots, s_{|V|-2} = s_{n-1})$ of D-,S- or P-operations that constructs the graph G from K_2. Now let $G' = (V', E')$ be the DSP-graph constructed by the sequence $(s_1, \ldots, s_{n-2})$. Since G' has n vertices, the treewidth of G' is less than or equal to 2. So there is a tree decomposition $T = (B, I)$ of G' with width 2. Now we will show that T can be extended to form a tree decomposition of G with width of 2. Since G can be constructed by applying the operation s_{n-2} on G', we will do a case distinction on s_{n-2}:

- If s_{n-2} is a D-operation adding the leaf $c \notin V \setminus V'$ to a vertex $v \in V'$, T can be extended by a bag $\{c, v\}$ that is connected to the subtree of v in T.
- If s_{n-2} is a S- or a P-operation adding a vertex c to an edge $\{a, b\} \in E'$, T can be extended as follows: since $\{a, b\}$ is a clique, there is a bag W with $\{a, b\} \subseteq W$. If we add the bag $\{a, b, c\}$ and connect it to W, T is still a tree decomposition with width 2.

□

Now we show that the MinDegree heuristic always works optimally on partial 2-trees. For that, we show that every partial 2-tree has a minimum degree of at most 2, and that the partial 2-trees are closed under all eliminations with a degree of at most 2. Actually, we show a slightly stronger property of DSP-graphs in Lemma 5.6, which helps us in the subsequent lemmas.

Lemma 5.6. *If G is a DSP-graph, then G has at least two vertices with a degree of at most 2. If G is not complete there are two such vertices that are not adjacent.*

Proof. At first, we prove the second part by induction over the number of vertices in $G = (V, E)$. To make sure that G is not complete the base case in this induction starts with $|V| = 4$, since the K_4 is the forbidden minor of the DSP-graphs. The base case is illustrated in Figure 5.1.

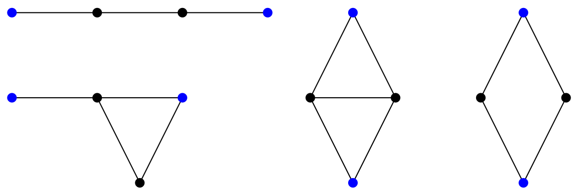

Figure 5.1: All DSP-graphs with four vertices. For each one of them, two vertices with a degree of at most 2 are colored in blue.

For the induction step let $o \in \{D, S, P\}$ be the last operation in the construction of G, v the vertex that was added during o, and G' the DSP-graph G before o was performed. By induction hypotheses, we know that G' has at least two non-adjacent vertices $u, w \in V$ with a degree of at most 2. If we apply o to G' the degree of at most one of those vertices can change, because o only affects one vertex (D) or two adjacent vertices (S and P). Let u be the vertex that is not affected by o. Since each possible operation o creates a new vertex v with a degree of at most 2, u and v are two non-adjacent vertices with a degree of at most 2.

As a result, we proved the second statement in the lemma, and therefore also the first one, for every DSP-graph with at least four vertices. For every other DSP-graph the lemma holds too, because: if $|V| \leq 3$, G can only be K_2,K_3 or a path with length 3. Since all those graphs have a maximum degree of 2, the lemma now holds for every DSP-graph. □

Lemma 5.7. *Let $G = (V, E)$ be a partial 2-tree and let G' be the elimination graph of G with respect to a vertex $v \in V$ with $\deg(v) \leq 2$. Then G' is a partial 2-tree.*

Proof. Lets assume that G' is not a partial 2-tree, i.e., $K_4 \preccurlyeq G'$. Because G' is obtained by an elimination of a vertex v with $\deg(v) \leq 2$ and all possible elimination steps of v create minors of G, G' is a minor of G. Since $K_4 \preccurlyeq G' \preccurlyeq G$, the graph G is not a partial 2-tree, which is a contradiction. □

Theorem 5.8. *The* MinDegree *heuristic returns the correct treewidth for every partial 2-tree.*

Proof. We know that the MinDegree heuristic returns the correct treewidth for trees (Theorem 5.3). So it remains to show that the MinDegree heuristic returns an upper bound of at most 2 for every partial 2-tree. This is done by Lemma 5.6 and Lemma 5.7. □

5.3 From Grid to Grill

Using the findings from the previous section, we show the behavior of the MinDegree heuristic on a more general form of grids that is based on partial 2-trees. This kind of graph, called *grill*, was mentioned by Sergey Norin in his lecture about minor theory during the proof of the *grid theorem* [Nor17, Theorem 6.1]. This theorem states that for every $n \in \mathbb{N}$ it exists an $N \in \mathbb{N}$ such that all graphs G with $tw(G) \geq N$ have an $n{\times}n$-grid as minor. The idea of the proof is to show step by step that such a graph G contains a hierarchy of increasingly structured but smaller subgraphs, the most grid-like of which is the grill. Informally, a grill is a grid, in which every edge can be replaced by a whole path and all outer vertices may have an additional path as shown in Figure 5.2.

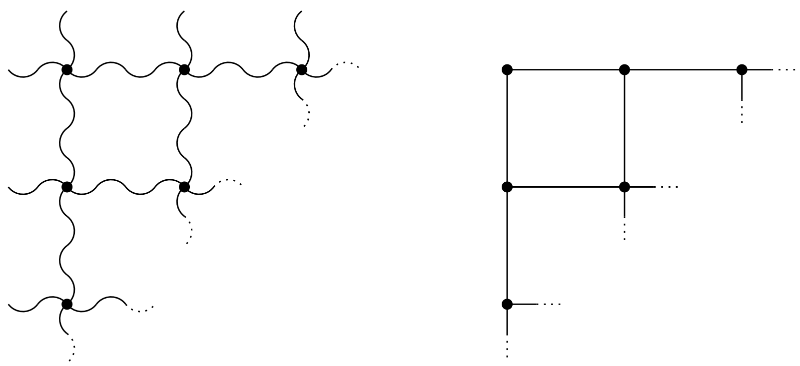

Figure 5.2: The general structure of a grill (left) in comparison to a general grid (right). The curly lines represent whole paths.

In this section, we show that the MinDegree heuristic returns the same result on an $n \times m$-grill as on a $n{\times}m$-grid. In fact, we show this statement for a superclass of grills, in which every edge of a grill can be replaced by a whole *series-parallel graph*, i.e., a kind of partial 2-tree. We call the elements of this superclass *fat $n \times m$-grills*, since these series-parallel graphs, which are defined in the following, can be seen as bloated paths.

Definition 5.9 ([Bod98])**.** An *s,t-series-parallel graph* $G = (V, E, s, t)$ is a graph $G = (V, E)$ with two specific vertices $s \in V$ and $t \in V$ called *source* and *sink* respectively. This graph class is characterized by the following constructive definition:

- $G = (\{s, t\}, \{\{s, t\}\}, s, t)$ is a series-parallel graph
- Ff $G_1 = (V_1, E_1, s_1, t_1)$ and $G_2 = (V_2, E_2, s_2, t_2)$ are series-parallel graphs, then:
  - the *parallel composition* G_1 and G_2 is a series-parallel graph, which can be constructed by the following steps: take the disjoint union of G_1 and G_2, add the edges $\{s_1, s_2\}$ and $\{t_1, t_2\}$ and contract them afterwards. The so created vertices of these edges are the source and sink of the remaining graph respectively.
  - the *series composition* of G_1 and G_2 is series-parallel, which can be constructed by the following steps: take the disjoint union of G_1 and G_2, add the edge $\{t_1, s_2\}$ and contract it. Then s_1 is the new source, and t_2 the new sink.

A graph $G = (V, E)$ is called *series-parallel* if there are vertices $s, t \in V$ such that (V, E, s, t) is a series-parallel-graph.

It is well known that every s,t-series-parallel graph is a partial 2-tree [Bod98, Theorem 41]. Based on this fact, we transfer the ideas about partial 2-trees made in Section 5.2 to s,t-series-parallel graph. We know that the MinDegree heuristic always returns the correct value for series-parallel graph (Theorem 5.8). In the following, we show that the MinDegree heuristic does the same, even if the vertices s and t are somehow blocked, i.e., have a higher degree as in fat grills (Figure 5.3).

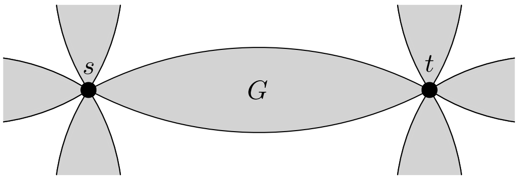

Figure 5.3: A cutout from a fat grill. Each of the grey areas represents a series-parallel graph.

Lemma 5.10. *If $G = (V, E, s, t)$ is a series-parallel subgraph of a graph in which s and t have at least degree 3. Then the* MinDegree *heuristic will reduce G to a single edge between s and t without eliminating a vertex with a higher degree than 2.*

Proof. The key part of this proof is to show that every s,t-series-parallel graph $G = (V, E)$ with at least 3 vertices has at least one vertex $v \in V \setminus \{s, t\}$ with $\deg(v) \leq 2$. We prove this claim by induction over the composition steps of an $s - t$-series-parallel graph. In the base case, every graph with at most three vertices has a maximum degree of 2. Therefore it either has less than three vertices or a vertex $v \in V \setminus \{s, t\}$ with $\deg(v) \leq 2$. Now let G_1 and G_2 be two s,t-series-parallel graphs. We show that both the series-composition as well as the parallel composition of G_1 and G_2 have a vertex with a degree of at least 2 besides their dedicated source and sink vertices. If one of the graphs G_1 and G_2 has more than three vertices, there is at least one vertex v with a degree of at most 2 whose degree is not changed during any of these compositions. Otherwise, if both G_1 and G_2 only have two vertices, a parallel composition does not increase the number of vertices, and a series composition results in a path with length 3. Since we know from the base case that this path has a third vertex with a degree of at most 2, the proof of this claim is finished.

Using this claim, we can show that the MinDegree heuristic eliminates only vertices with a degree of at most 2, until only an edge between s and t remains. If G has more than two vertices, there is at least one vertex with a degree of at most 2. Therefore there is vertex v with minimum degree besides s and t. If we eliminate this vertex, the resulting graph is again a series-parallel graph (Lemma 5.7), which is why we can use the same argumentation in all the following steps until G has only two vertices, i.e., s and t, left. Since s and t were connected (but not necessarily adjacent), they are adjacent in the resulting graph, which proves this lemma. □

Based on this Lemma 5.10 and the fact that only vertices with degree 2 and smaller are eliminated, we can say that the MinDegree heuristic reduces the series-parallel subgraphs of a graph first without eliminating a vertex with a degree higher than 2. As a result, a fat $n \times m$-grill is reduced to an $n{\times}m$-grid where the four corner vertices are already eliminated (since they are series-parallel subgraphs themselves in a grid). The transition from a fat $n \times n$-grill to a grid with eliminated corners is shown in Figure 5.4.

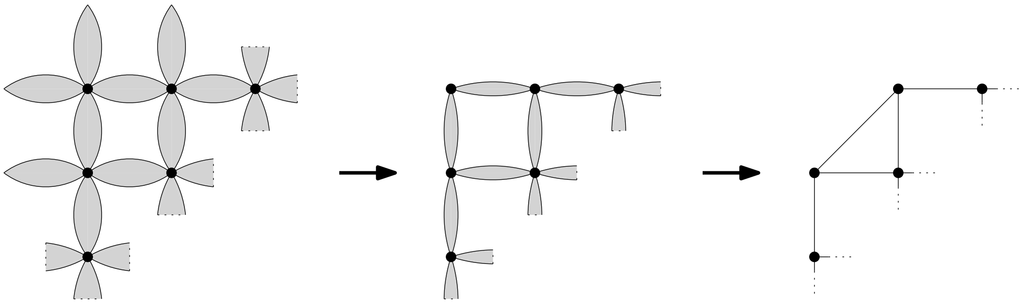

Figure 5.4: The transition from a fat grill (left) to a grid with eliminated corners (right). The grey areas are series-parallel graphs. The figure in the middle shows the state where all outer series-parallel graphs are completely eliminated, and inner series-parallel graphs remain, that are perhaps a bit thinner.

From this point on, the MinDegree heuristic behaves the same as on grids, because an $n{\times}m$-grid is also reduced to such a grid with rounded corners first. Consequently, the result for a fat $n \times m$-grill is the maximum of the results of an $n{\times}m$-grid and the series-parallel parts of the grill.

5.4 3xn-Grids

In this section we proof that the MinDegree heuristic returns the correct treewidth of 3 for every $3{\times}n$-grid with $n \in \mathbb{N}$. For this, we show that this statement holds for the following graph class $\mathcal{G}_3$ that includes all $3{\times}n$-grids.

Definition 5.11. A graph $G = (V, E)$ is a member of $\mathcal{G}_3$ if:

(1) there is a subset of vertices $C \subseteq V$ such that
- $G[C]$ is a chordless cycle with length $|C| \geq 0$ and
- $P \coloneqq V \setminus C$ induces a path with length $|P| \geq 0$ in G, and

(2) G is connected, and

(3) G is planar

Note that we still call C a circle, if $|C|$ is 1 or 2. In this case, C is only one vertex or two vertices that are connected by an edge, respectively.

A structure of a graph $G \in \mathcal{G}_3$ is shown in Figure 5.5. Note that every $3{\times}n$-grid with $n \geq 3$ is part of this graph class.

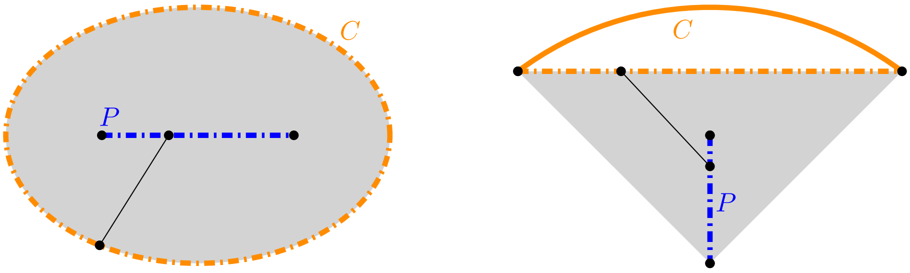

Figure 5.5: Two possible viewpoints on a graph in the graph class $\mathcal{G}_3$. The dotted-dashed lines are placeholders for vertices that are connected along these lines. Solid lines represent edges. An arbitrary number of edges connecting P and C can be located in the grey area. One of those edges has to exist if P and C are not empty to ensure that the graph is connected.

To show that the MinDegree heuristic returns a value of at most 3 for all graphs in $\mathcal{G}_3$, we prove that a graph in $\mathcal{G}_3$ has a minimum degree of at most 3, and that $\mathcal{G}_3$ is closed under matching eliminations.

Lemma 5.12. *If $G \in \mathcal{G}_3$ then $\delta(G) \leq 3$.*

Proof. Let $G = (V, E)$ be a graph in $\mathcal{G}_3$ with cycle C and path P. If $|C| = 0$, then G is only a path where the minimum degree is at most 1. If $|P| = 0$, then G is only a chordless cycle, in which the minimum degree is at most 2. Otherwise, let $v \in V$ be a leaf in $G[P]$. If we assume $\delta(G) > 3$, then v has at least three neighbors in C. Now let $c_1, c_2, c_3 \in C$ be those neighbors ordered clockwise by the angle in which their respective edge enters the vertex v. An illustration of this situation is shown in Figure 5.6.

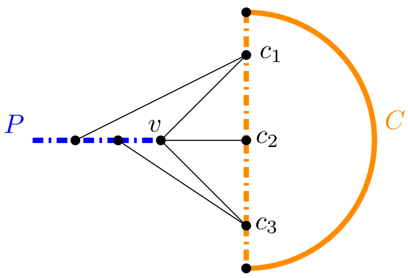

Figure 5.6: Illustration of the neighborhood of v in a graph of $\mathcal{G}_2$.

By assumption c_1 and c_3 have a degree of at least 4. The problem is that c_2 can not be adjacent to another vertex anymore because:

- If c_2 was adjacent to a vertex in C, $G[C]$ would not be a chordless cycle.
- If c_2 was adjacent to another vertex in P, the edge had to cross at least one other edge.

As a consequence, there is a vertex with degree less than the minimum degree, which is a contradiction. □

Next, we show that $\mathcal{G}_3$ is closed under all eliminations with degree of at most 3 that can occur during the MinDegree heuristic. To prove that planarity remains after eliminating a vertex in a graph $G \in \mathcal{G}_3$, we introduce the following class of eliminations. Let $G = (V, E)$ be a planar graph, $v \in V$ a vertex and G' their elimination graph. Then the elimination of v is called a *planar elimination* if G' is also a planar graph. In Lemma 5.13 we show that all eliminations possible for a $\mathcal{G}_3$ graph remain planarity, and therefore property (3) of the Definition 5.11.

Lemma 5.13. *All eliminations with degree of at most* 3 *are planar eliminations.*

Proof. Most eliminations with a degree at most 3 are planar eliminations by the fact that planar graphs are closed under taking the minor. For that, note that the elimination graph of a graph G with respect to an almost simplicial vertex is a minor of G. The reason for that is, that the elimination of an almost simplicial vertex can be simulated by contracting the edge between the eliminated vertex and its non-clique vertex. Since all eliminations with degree of at most 3 and fill-in of at most 2 eliminate an almost simplicial vertex, these eliminations are planar eliminations. Consequently, it remains to show that $(3, 3)$-eliminations remain planarity too. Let $G = (V, E)$ be a planar graph and let $v \in V$ be a vertex whose elimination is a $(3, 3)$-elimination. Because G is planar, the edges from v to its neighbors do not take up space that is taken by other edges. This space can be used to connect the neighbors of v to a 3-clique in the elimination graph of G and v, as shown in Figure 5.7.

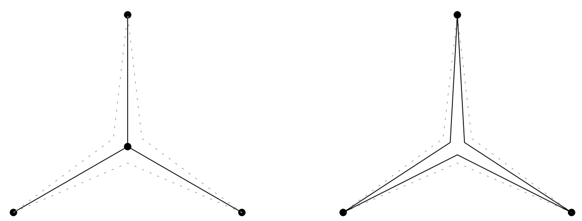

Figure 5.7: Illustration for a planar $(3, 3)$-elimination.

□

That the elimination graph of a graph in $\mathcal{G}_3$ is still connected is shown in Lemma 4.1. Therefore it remains to show that the elimination graph G' of a graph $G \in \mathcal{G}_3$ with respect to a vertex with minimum degree can also be partitioned into a chordless cycle and a path (property (1) in Definition 5.11 of $\mathcal{G}_3$). We proof this in Lemmas 5.16 to 5.18 separated by the degree of the elimination, after showing some helpful properties in Corollary 5.14 and Lemma 5.15.

Corollary 5.14 (from Definition 5.11)**.** *Let $G = (V, E) \in \mathcal{G}_3$ with dedicated cycle $C \subseteq V$ and path $P \subseteq V$. Further, let G' be the graph G after adding or removing arbitrary CP–edges. Then $G' \in \mathcal{G}_3$ if G' is planar and connected.*

Proof. Adding or removing CP–edges does neither affect the sets C and P, nor the edges in $G[P]$ and $G[C]$. Therefore C and P remain a cycle and a path in G' respectively. Since G' is planar and connected by assumption, all conditions for $G' \in \mathcal{G}_3$ are fulfilled. □

Lemma 5.15. *Let $G = (V, E) \in \mathcal{G}_3$ with dedicated cycle $C \subseteq V$ and path $P \subseteq V$, and let G' be the graph G after contracting an P– or C–edge. Then G' is in $\mathcal{G}_3$.*

Proof. The graph G' is planar, because G is planar, and G' is by definition a minor of G. It is also connected by Lemma 4.2. Let $e = \{x, y\} \in E$ be an edge in G, and z the vertex e

is contracted to. If $x, y \in C$, then $G[P]$ is not changed at all during the contraction of e, and $(C \cup \{z\}) \setminus \{x, y\}$ induces a cycle in G'. This cycle is additionally chordless, because G would otherwise have a chord too. If $x, y \in P$, then $G[C]$ is not changed at all during the contraction of e, and $(P \cup \{z\}) \setminus \{x, y\}$ induces a path in G'. Therefore G' fulfills all conditions for $\mathcal{G}_3$. $\square$

With those properties of $\mathcal{G}_3$, we prove for every elimination with minimum degree, that the elimination graph is still in $\mathcal{G}_3$. Since the minimum degree in a graph of $\mathcal{G}_3$ is 3 (Lemma 5.12), it is sufficient to show this for eliminations with degree of at most 3. In the following lemmas, we consider the possible eliminations separately depending on their degree.

Lemma 5.16. *The graph class $\mathcal{G}_3$ is closed under eliminating leaves.*

Proof. Let $G = (V, E)$ be a graph in $\mathcal{G}_3$ with dedicated cycle $C \subseteq V$ and path $P \subseteq V$, $v \in V$ a leaf in G, and G' the elimination graph of G with respect to v. Because v is a leaf it has only one neighbor $w \in V$. Now there are the following two cases. If v and w are both in $G[C]$ (or $G[P]$), then $G' \in \mathcal{G}_3$ according to Lemma 5.15. If, otherwise, $v \in P$ and $w \in C$, then v is the last vertex in P and $\{v, w\}$ is the only edge connecting P and C. Therefore $G' = G[C]$, which is planar, connected and has a chordless cycle C and a path $P = \emptyset$. So $G' \in \mathcal{G}_3$ applies. For $v \in C$ and $w \in P$ the proof is analog. $\square$

Lemma 5.17. *Let $G = (V, E)$ be a graph in $\mathcal{G}_3$ with dedicated cycle $C \subseteq V$ and path $P \subseteq V$. Further, let $\delta(G) = 2$ and let $v \in V$ be a vertex with minimum degree. Then G', the elimination graph of G respect to v, is in $\mathcal{G}_3$.*

Proof. The graph G' is connected and planar by Lemmas 4.2 and 5.13 respectively. Let $a, b \in V$ be the two neighbors of v. Since a vertex with degree 2 is always an almost simplicial vertex, the elimination of v can be simulated by either contracting $\{v, a\}$ or $\{v, b\}$. If one of those edges is either a P- or a C–edge, then $G' \in \mathcal{G}_3$ according to Lemma 5.15.

If otherwise both edges are CP–edges, we show that the vertices of G can be partitioned into a path P' and a cycle C', such that v is incident to a C'–edge. Then we can use Lemma 5.15 again to show $G' \in \mathcal{G}_3$. Both cases are illustrated in Figure 5.8. If $v \in P$, then v have to be the only vertex in P, because it is not incident to a P–edge by assumption. Now, let $a, b \in C$ the two neighbors of V. Because a and b both lie on the cycle, they are connected by two path S_1 and S_2. One of those paths, w.l.o.g. S_1, forms a cycle C' together with the edges $\{v, a\}$ and $\{v, b\}$, while the other one remains a path. If v is a vertex in C and $a, b \in P$ the two neighbors of v, then v is the only vertex in C. Since we assumed a minimum degree of 2, a and b have to be the leafs of $G[P]$. Therefore the whole graph is a single cycle with vertices $C' = V$.

$\square$

Lemma 5.18. *Let $G = (V, E)$ be a graph in $\mathcal{G}_3$ with dedicated cycle $C \subseteq V$ and path $P \subseteq V$. Further, let $\delta(G) = 3$ and let $v \in V$ be a vertex with minimum degree. Then G', the elimination graph of G respect to v, is in $\mathcal{G}_3$.*

Proof. Again, G' is connected and planar by Lemmas 4.2 and 5.13 respectively. To proof that G' also fulfills the third property of Definition 5.11, we show that all possible eliminations of v can be simulated by a sequence of operations described in Lemma 5.15 and Corollary 5.14. For this, we make a case distinction over the position of v in G, i.e., we consider the situations where v is a vertex in the cycle or the path.

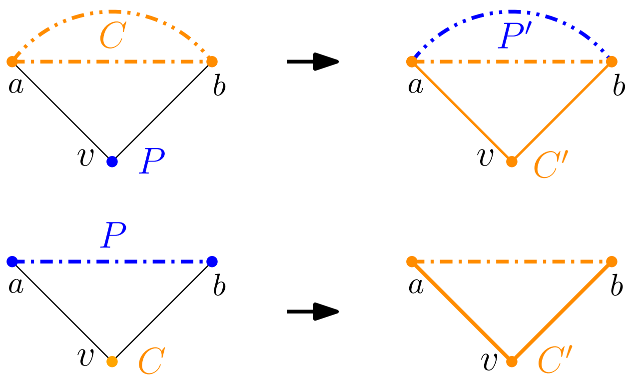

Figure 5.8: Proof that the vertices of G can be partitioned into a path P' and a cycle C', such that v is incident to a C'–edge.

Let v be a vertex in the cycle C. In the following we make a case distinction over the size of C: If $|C| \geq 3$, v has exactly two neighbors $a, b \in C$ and, because of $\delta(G) = 3$, a neighbor $c \in P$. In this case, a single edge contraction is not always sufficient to simulate the elimination of v. After the elimination of v, the edges $\{a, b\}$, $\{a, c\}$ and $\{b, c\}$ have to exist in G, while contracting the C–edge $\{v, a\}$ (or $\{v, b\}$) in G only ensures the edges $\{a, b\}$ & $\{v, a\}$ (or $\{a, b\}$ & $\{v, b\}$) to exist afterwards. In both cases the only missing edges are CP–edges, which can safely be added without loosing the properties of $\mathcal{G}_3$, according to Corollary 5.14.

If $v \in C$ with $|C| = 2$, then the vertices of G can be partitioned in another way, such that there is a path P' with $|P'| \geq 0$ and a cycle $C' := V \setminus P'$ with $|C'| \geq 3$ and $v \in C'$. As a result, one of the cases above can be applied. Note that in this case both leafs of $G[P]$ have to be adjacent to both vertices in C to ensure the minimum degree of 3. If C had only one vertex there would not be enough CP–edges for the leafs of P to have degree 3. Since this would imply the existence of a vertex with degree of at most 2, this is a contradiction to $\delta(G) = 3$. Both cases for $|C| \leq 2$ are illustrated in Figure 5.9.

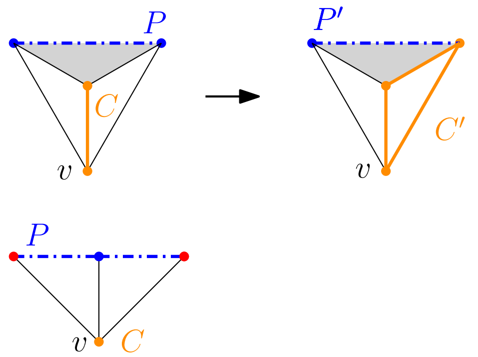

Figure 5.9: Illustration for $\delta(G) = 3$ and $|C| = 2$ (upper figure) and $|C| = 1$ (lower figure). The gray area in the upper case is a placeholder for any number of CP–edges. The red vertices in the lower figure have to have a degree of 2, which is a contradiction in this case.

Now, let v be a vertex on the path P of G. In the following, we make a case distinction over the degree of v in $G[P]$. Note that this degree can not be greater than 2, because otherwise $G[P]$ was not a path.

If v has a degree of 2 in $G[P]$, then v has exactly two neighbors $a, b \in P$ and one neighbor $c \in C$. Now the argumentation is the same as in the case „$|C| \geq 3$“.

If, otherwise, v has a degree of 1 in $G[P]$, let $w \in V$ be the only neighbor of v in $G[P]$. In this case, we can show that contracting the edge $\{v, w\}$ simulates all possible eliminations of v. As this is an operation according to Lemma 5.15, we can conclude $G' \in \mathcal{G}_3$. For this, we show that w is always an almost-clique-vertex of v. If $a, b \in C$ are the other neighbors of v in $G[C]$, we have to show that they are adjacent. So, assume they are not adjacent. Then there is at least one vertex $m \in C$ that lies between a and b on the cycle $G[C]$. This vertex m can not be adjacent to any other vertex as its two neighbors in C, because all three neighbors of v already mentioned, and any other edge would be a chord in $G[C]$. Therefore m is a vertex with degree 2, which is a contradiction to v being a vertex with minimum degree. As a consequence, a and b are adjacent and therefore also a clique with size $2 = \deg(v) - 1$. This implies that w is an almost-clique-vertex of v, which is why contracting the edge $\{v, w\}$ simulates all possible eliminations of v in this case.

If $v \in P$ has a degree of 0 in P, v is the only vertex in P. With the same argumentation we can show that $G = K_4$ is the only graph in $\mathcal{G}_3$ meeting these conditions. In this case we show that the vertices of G can be partitioned in another way, such that there is path P' with $|P'| \geq 0$ and a cycle $C' := V \setminus P'$ with $|C'| \geq 3$ that includes v. Therefore a subcase in „$|C| \geq 3$“ can be applied. This situation is shown in Figure 5.10.

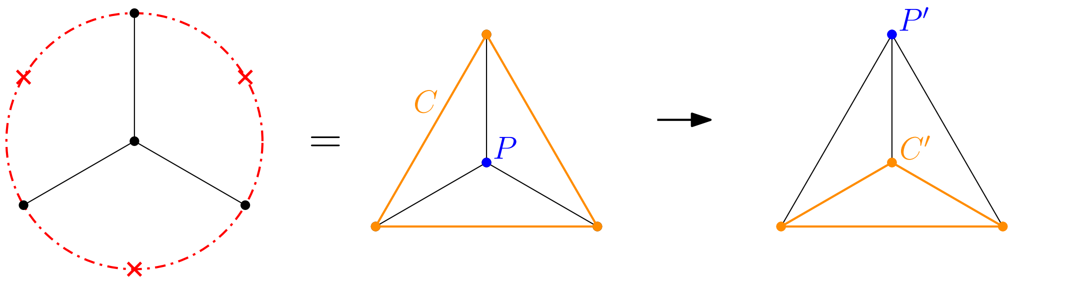

Figure 5.10: The red vertices and edges in the left graph show that there can not be any vertices between the neighbors of the center vertex v. As a result, the graph in the middle is the only one of this kind. In this graph, there is a cycle C' with at least three vertices that includes the center vertex v. This cycle is shown in the right graph.

□

With those lemmas, we can prove the following Theorem 5.19.

Theorem 5.19. *The* MinDegree *heuristic returns the correct treewidth for every* $3{\times}n$*-grid.*

Proof. In Lemma 5.12 and the Lemmas 5.16 to 5.18 we prove the necessary conditions to show that the MinDegree heuristic returns a value of at most 3 for every graph in $\mathcal{G}_3$. Since each $3{\times}n$-grid is a member of $\mathcal{G}_3$, this applies to every $3{\times}n$-grid too. From Theorem 5.8 we know that the MinDegree heuristic returns the correct treewidth for the $3{\times}1$-grid and the $3{\times}2$-grid. For all other $3{\times}n$-grids the MinDegree heuristic returns a value that is at most the actual value. Since this heuristic is an upper-bound heuristic, the theorem holds for every $3{\times}n$-grid. □

6. Approaches for MinDegree on Larger Grids

This chapter is about finding an upper bound to the results of the MinDegree heuristic on *larger grids*, i.e., $n \times m$-grids with $n, m \geq 4$. For this, we show in Section 6.1 why the size of the *first clique*, that occurs in the sequence of elimination graphs of the heuristic, is so important. Using these findings, we present two approaches in Sections 6.2 and 6.3 to find such upper bounds.

6.1 The First Clique

In Theorem 6.3 we show how the size of the first clique and the heuristic result are related to each other. For this, we first consider chordal graphs in Lemma 6.1 and then extend the statement to general graphs.

In the proofs of the following lemmas, we argue with the sequence of elimination graphs of a graph $G = (V, E)$ in the course of the MinDegree heuristic. These graphs are denoted as $G = G_0, G_1, \dots, G_{|V|}$, where G_i is the graph G after the i-th iteration of the heuristic. In the next lemma, we show an interesting property of the graph G_k, where k is the highest number such that G_k has minimum degree $t = \text{MinDegree}(G)_{\text{tw}}$. This k exists, because there has to be at least one graph with the minimum degree t in the sequence $G_1, \dots, G_{|V|}$ if the MinDegree heuristic returns the value t.

Lemma 6.1. *Let $G = (V, E)$ be a chordal graph and G_i the graph G after the i-th iteration of the* MinDegree *heuristic. Further, let G_k with $k = 1, \dots, |V|$ be the last graph with a minimum degree of* MinDegree$(G)_{tw}$ *in the sequence $G_1, \dots, G_{|V|}$. Then G_k is a clique with size $tw(G) + 1$.*

Proof. Assume $G_k = (V, E)$ is not a clique. Because G is chordal and chordal graphs are closed under eliminations according to Corollary 4.4, G_k is also a chordal graph. Since G_k is not complete, G_k has at least two non-adjacent simplicial vertices u and v [Jr.83]. In the following, we show that, under these assumptions, the elimination graph of G_k with respect to an arbitrary vertex x with minimum degree has a clique with size $\text{tw}(G) + 1$. Therefore the MinDegree heuristic is forced to eliminate at least one vertex with degree $\text{tw}(G)$ at some point, which is a contradiction to G_k being the last graph in the sequence with that minimum degree.

First, we show that the degree of u and v is exactly $\mathrm{tw}(G)$. If we assume that they have a higher degree than $\mathrm{tw}(G)$, there is a clique in G_k with a size of at least $\mathrm{tw}(G) + 2$. This implies that G_k has a treewidth of at last $\mathrm{tw}(G) + 1$. Therefore the MinDegree heuristic would return a higher value than the actual treewith of G if G_k is in the sequence. Since the MinDegree heuristic works optimal on chordal graphs (Theorem 5.3), G_k can not be part of the sequence $G_1, \ldots, G_{|V|}$, which is a contradiction. Therefore $\deg(u), \deg(v) \leq \mathrm{tw}(G)$ applies. Additionally, u and v have a degree of at least $\mathrm{tw}(G)$ because G_k has a minimum degree of $\mathrm{tw}(G)$.

Let x be a vertex with minimum degree in G_k (not necessarily u or v). In the following we show that the elimination graph of G_k with respect to x still has a clique with size $\mathrm{tw}(G) + 1$. For this, we show that the elimination of x can only change the degree of at most one of the vertices u or v. If $x \in \{u, v\}$, w.l.o.g $x = u$, then the elimination of x does not change the neighborhood of v, because u and v are not adjacent (but simplicial). If otherwise $x \notin \{u, v\}$, it remains to show that x is not in the neighborhood of both u and v. So assume x was a neighbor of u and v. Since v is simplicial in G_k, x is also adjacent to all neighbors of v. Therefore x has at least $\mathrm{tw}(G)$ neighbors. But because x is also adjacent to u, and $u \notin N_{G_k}(v)$, the vertex x even has at last $\mathrm{tw}(G) + 1$ neighbors. This is a contradiction to x being a vertex with minimum degree.

As a result, G_{k+1}, the elimination graph of G_k with respect to x, still has a simplicial vertex with degree $\mathrm{tw}(G)$. Therefore there is a clique of size $\mathrm{tw}(G) + 1$ in G_{k+1}, which forces the MinDegree heuristic to eliminate at least one vertex with degree $\mathrm{tw}(G)$ at some point. This is a contradiction to G_k being the last graph in which such a vertex is eliminated. As a consequence, the assumption that G_k is not complete was wrong. Since G_k has a minimum degree of $\textsc{MinDegree}(G)_{\mathrm{tw}} = \mathrm{tw}(G)$ by assumption, G_k is a clique with size $\mathrm{tw}(G) + 1$. □

In the following, we extend the idea to general graphs. For this, we show the connection of a general graph G and one of its triangulations in Lemma 6.2.

Lemma 6.2. *Let $G = (V, E)$ be a graph and $G^\Delta = (V, E^\Delta)$ the triangulation of G according to* $\textsc{MinDegree}(G)_\pi$. *Then* $\textsc{MinDegree}(G, \rho)$ *and* $\textsc{MinDegree}(G^\Delta, \rho)$ *have the same elimination ordering and the same sequence of degrees.*

Proof. For this proof, we show that $\textsc{MinDegree}(G, \rho)$ and $\textsc{MinDegree}(G^\Delta, \rho)$ eliminate the same vertex $v \in V$ with the same degree in the first step. Each further step works with the same arguments if we replace G with the elimination graph of G and v. Let v be the first vertex that is eliminated in $\textsc{MinDegree}(G, \rho)$, i.e., the vertex with minimum degree that has the lowest value in $g(v)$. Since v was eliminated first, there cannot be any additional fill-in edges in G^Δ that are incident to v. Therefore v has the same degree in G^Δ as in G. Since all other vertices in G^Δ have at least the degree that they have in G, v is also eliminated first in $\textsc{MinDegree}(G^\Delta, \rho)$. □

Theorem 6.3. *Let $G = (V, E)$ be a graph, ρ a vertex ordering of G, and $G_0 = G, G_1, \ldots, G_{|V|}$ the sequence of elimination graphs of the* MinDegree *heuristic with ρ. Then the first complete elimination graph in this sequence has the index $k = |V| - (\textsc{MinDegree}(G, \rho)_{tw} + 1)$.*

Proof. For reasons of clarity let $t = \textsc{MinDegree}(G, \rho)_{\mathrm{tw}}$. Further, let k be the index of the first complete graph in the sequence $G_1, \ldots, G_{|V|}$. We show that $k = |V| - (t + 1)$ by leading both $k < |V| - (t + 1)$ and $k > |V| - (t + 1)$ to a contradiction.

Assume $k < |V| - (t + 1)$. In general the graph G_i with $i \in [1; |V|]$ has $|V| - i$ vertices. Therefore the graph G_k has strictly more than $t + 1$ vertices. Since G_k is a complete graph, the minimum degree of G_k is strictly greater than t. As a consequence, the MinDegree heuristic eliminates a vertex with a degree of at least $t+1$ next, which is a contradiction to the definition of $t = \text{MinDegree}(G,\rho)_{\text{tw}}$ being the highest degree of eliminated vertices in the sequence $G_1, \ldots, G_{|V|}$.

To prove $k > |V|-(t+1)$ wrong, let G^Δ be the triangulation of G according to the elimination ordering $\text{MinDegree}(G,\rho)_\pi$, and let $G_1^\Delta, \ldots, G_{|V|}^\Delta$ be the sequence of elimination graphs of the MinDegree heuristic with ρ. Then $\text{MinDegree}(G,\rho)$ and $\text{MinDegree}(G^\Delta,\rho)$ have the same elimination ordering and the same sequence of degrees (Lemma 6.2). Now assume that the first complete graph G_k has less than $t+1$ vertices, i.e., $k > |V| - (t+1)$. In the following we lead this to a contradiction by finding an $l < k$ such that G_l is already a complete graph, which is a contradiction to the definition of k. Since G_k has by assumption less than $t+1$ vertices, its minimum degree is strictly less than t. Therefore there has to be a graph earlier in the sequence $G_1, \ldots, G_{|V|}$ that has a minimum degree of $t = \text{MinDegree}(G,\rho)_{\text{tw}}$. Let l be highest index with $l < k$, such that the minimum degree of G_l is t. Because G and G^Δ have the same degree sequence of eliminated vertices (Lemma 6.2), G_l and G_l^Δ have the same minimal degree, namely $\text{MinDegree}(G,\rho)_{\text{tw}}$. As G_l^Δ has minimal degree $t = \text{MinDegree}(G,\rho)_{\text{tw}}$, it is complete (Lemma 6.1). As the graph G_l has as many vertices as G_l^Δ, namely $t+1$, and has a minimum degree of t, it has to have at least $(t+1)t/2$ edges, making it complete. □

As a result, the key idea to get an upper bound to the result of the MinDegree on an arbitrary graph is to bound the size of the first clique in an elimination sequence from above. Different approaches to that idea are presented in the following two sections.

6.2 Counting argument

The first approach ignores the structure of a grid and focuses only on the distribution of all degrees. We assume that all vertices have the same degree, i.e., the average degree of the graph, to raise the minimum degree. In the following, we present a procedure to gradually increase the average degree by eliminations with maximal fill-in. For that, we describe this procedure for the first step and then generalize the idea, that leads to Theorem 6.6, in which we give an upper bound to the result of the MinDegree heuristic on $n{\times}n$-grids.

An $n{\times}n$-grid has $|V| = n^2$ vertices and $|E| = 2n^2 - 2n$ edges, so the average degree is

$$2\frac{|E|}{|V|} = 2\frac{2n^2-2n}{n^2} = \frac{4n^2}{n^2} - \frac{4n}{n^2} < 4.$$

As the average degree is strictly less than 4, there has to be a vertex with a degree of at most 3. Actually, there are even four vertices with a degree of 2, but as we search for an upper bound, we take the highest possible value for the minimum degree. In the next step, we eliminate as many vertices with degree 3 as necessary to reach an average degree of 4. For that, let k be the number of such vertices. It is easy to see that the number of vertices decreases by k during these eliminations. To see how the number of edges changes, we consider the fill-in of all possible elimination with degree 3, and find an upper bound. In an elimination with degree 3 at most $3 * (3-1)/2 = 3$ edges can be added in between the neighbors of the eliminated vertex. Since the three edges from the eliminated vertex to its neighbors are removed by that, the edge difference is at most 0. To increase the average degree to 4, k has to fulfill

$$
\begin{aligned}
4 &\leq 2\frac{|E'|}{|V'|} = \frac{|E| + 0 \cdot k}{|V| - k} = 2\frac{2n^2 - 2n}{n^2 - k} \\
\Leftrightarrow &\ 4n^2 - 4k \leq 4n^2 - 4n \\
\Leftrightarrow &\ k \geq n.
\end{aligned}
$$

Consequently, after at least n eliminations the average degree rises to 4. After exactly n eliminations, there are $n^2 - n$ vertices and exactly twice as many edges. Since this term will occur frequently in this section, we define $N(n) = n^2 - n$.

Now, we check for which n there are more edges in the new graph than the number of vertices even allows. A graph with l vertices is complete if and only if it has $l(l-1)/2$ edges. If the new graph has more edges than possible, an elimination with degree 3 resulted in a complete graph. According to Theorem 6.3 the result of the MinDegree heuristic is then 3. Therefore the MinDegree heuristic returns at most 3 for all $n{\times}n$-grids such that

$$
\begin{aligned}
|E'| &\geq \frac{|V'|(|V'| - 1)}{2} \\
\Leftrightarrow &\ 4|V| \geq |V|(|V| - 1) \\
\Leftrightarrow &\ 4(n^2 - n) \geq (n^2 - n)(n^2 - n - 1) \\
\Leftrightarrow &\ 0 \leq -n^4 + 2n^3 + 4n^2 - 5n = x \cdot (x - 1) \cdot (-x^2 + x + 5) \\
\Leftrightarrow &\ n \in \left[\frac{-\sqrt{21} + 1}{1}; 0\right], \left[1; \frac{\sqrt{21} + 1}{2}\right].
\end{aligned}
$$

As a result, 3 is an upper bound for all $n{\times}n$-grids with $n \leq (\sqrt{21} + 1)/2 < 3$, which is already shown by Theorem 5.8. To find upper bounds for larger grids we repeat this step gradually for all average degrees. So, in the next step we to increase the average degree to 5 by eliminations with degree 4. Since computing all steps one after another is elaborate, we generalize the ideas presented so far and find a closed formula for an upper bound of the MinDegree heuristic that only depends on n.

For that, we generalize the formula that we used to calculate the necessary number of eliminations to increase the average degree to the next higher integer. Let $d = 2|E|/|V| \in \mathbb{N}$ be the current average degree of an elimination graph of the $n{\times}n$-grid (V, E). Additionally, let $f(d)$ be the maximum edge difference between the current graph and an elimination graph with respect to a vertex with degree d. This edge difference is at most $d(d-1)/2 - d$, because at most a clique with size d is created and the d edges between the eliminated vertex and its neighbors are removed. In general we can find an $a \in [3/8, 1/2]$ such that $f(d) = ad^2 - d$ is an upper bound to edge difference of an elimination with degree d. The lower bound to a comes from the fact, that the maximum edge difference of an elimination with degree 4 is 2. Substituting those two numbers into the function f gives

$$
a \cdot 4^2 - 4 \geq 2 \ \Rightarrow\ a \geq \frac{6}{16} = \frac{3}{8}.
$$

Now, the general formula for k to increase the average degree to the next integer $d+1$ is

$$
2\frac{|E| - f \cdot k}{|V| - k} \geq d + 1 \ \Rightarrow\ k \geq \frac{(d+1)|V| - 2|E|}{2 \cdot f(d) + (d+1)}.
$$

Substituting this lower bound for k into the fraction results in an average degree of exactly $d+1$. Additionally, we get the following upper bound to the new number of vertices $|V'|$.

$$\begin{aligned}
|V'| &= |V| - k \\
&\leq |V| - \frac{(d+1)|V| - 2|E|}{2 \cdot f(d) + (d+1)} \\
&= |V| \left(1 - \frac{(d+1) - 2\frac{|E|}{|V|}}{2 \cdot f(d) + (d+1)}\right)
\end{aligned}$$

In the previous step of this procedure, we increased the average degree to exactly d. Therefore we can substitute d for $2|E|/|V|$.

$$\begin{aligned}
&= |V| \left(1 - \frac{(d+1) - d}{2 \cdot f(d) + (d+1)}\right) \\
&= |V| \left(1 - \frac{1}{2 \cdot f(d) + (d+1)}\right)
\end{aligned}$$

Using this estimation repeatedly, we get an upper bound to the number of remaining vertices $|V_d|$ after all eliminations with a degree smaller than d, for some $d \geq 3$, that only depends on $N(n)$ and the function f.

$$\begin{aligned}
|V_d| &\leq |V_{d-1}| \left(1 - \frac{1}{2 \cdot f(d) + (d+1)}\right) \\
&\leq |V_{d-2}| \left(1 - \frac{1}{2 \cdot f(d-1) + d}\right) \left(1 - \frac{1}{2 \cdot f(d) + (d+1)}\right) \\
&\dots \\
&\leq N(n) \cdot \underbrace{\prod_{i=4}^{d} \left(1 - \frac{1}{2 \cdot f(i) + (i+1)}\right)}_{:=\Pi_d}
\end{aligned}$$

To get the new number of edges $|E'|$ after all eliminations with a degree smaller than d, we use $d+1 = 2|E'|/|V'|$, and get

$$|E'| = \frac{d+1}{2}|V'|.$$

Now, we find the smallest $d \geq 4$, such that the number of edges $|E_d|$, that remains after all eliminations with a degree smaller than d, exceeds the maximum number of edges in a graph with $|V_d|$ vertices. This is the case if

$$\begin{aligned}
&|E_d| \geq \frac{|V_d|(|V_d| - 1)}{2} \\
\Leftrightarrow\ &\frac{d+1}{2} \cdot |V_d| \geq \frac{|V_d|(|V_d| - 1)}{2} \\
\Leftrightarrow\ &d + 1 \geq |V_d| - 1.
\end{aligned}$$

By substituting $N(n) \cdot \Pi_d$ for $|V_d|$, we increase the smallest d, than fulfills this inequality. But, since we just need to find an upper bound to the smallest d, we can find the smallest d, that fulfills the inequality

$$\begin{aligned} d + 1 &\geq N(n) \cdot \Pi_d - 1 \\ \Leftrightarrow N(n) &\leq \frac{d+2}{\Pi_d} \end{aligned}$$

instead. To prove that the smallest d that fulfills this new condition is in $\Omega(n^2)$, we need the following technical Lemma 6.4.

Lemma 6.4. *For every $d \geq 4$ and $a \in [3/8, 1/2]$, $\Pi_d \geq 1/d + 1/2$ applies.*

Proof. We prove this inequality by induction over d with an arbitrary but fixed $a \in [3/8, 1/2]$. In the base case let $d = 4$.

$$\begin{aligned} \Pi_4 - \left(\frac{1}{4} + \frac{1}{2}\right) &= \left(1 - \frac{1}{2 \cdot f(4) + (4+1)}\right) - \left(\frac{1}{4} + \frac{1}{2}\right) \\ &= \left(1 - \frac{1}{2 \cdot (16a - 4) + 5}\right) - \frac{3}{4} \\ &= \left(1 - \frac{1}{32a - 3}\right) - \frac{3}{4} \\ &= \frac{32a - 3 - 1}{32a - 3} - \frac{3}{4} = \frac{4(32a-4) - 3(32a-3)}{4(32a-3)} \\ &= \frac{32a - 7}{128a - 12} \end{aligned}$$

Since both numerator and denominator are positive for all $a \in [3/8, 1/2]$, the whole fraction is always positive. Therefore Π_4 is at least $1/4 + 1/2$. In the induction step, the following applies.

$$\begin{aligned} \Pi_{d+1} - \left(\frac{1}{d+1} + \frac{1}{2}\right) &= \prod_{i=4}^{d} \left(1 - \frac{1}{2 \cdot f(i) + (i+1)}\right) - \left(\frac{1}{d+1} + \frac{1}{2}\right) \\ &\geq \left(\frac{1}{d} + \frac{1}{2}\right)\left(1 - \frac{1}{2 \cdot f(d+1) + (d+2)}\right) - \left(\frac{1}{d+1} + \frac{1}{2}\right) \\ &= \left(\frac{1}{d} + \frac{1}{2}\right) - \left(\frac{1}{d+1} + \frac{1}{2}\right) - \left(\frac{1}{d} + \frac{1}{2}\right)\frac{1}{2 \cdot f(d+1) + (d+2)} \\ &= \left(\frac{1}{d} - \frac{1}{d+1}\right) - \frac{2+d}{2d} \cdot \frac{1}{2 \cdot (a(d+1)^2 - (d+1)) + (d+2)} \\ &= \frac{1}{d(d+1)} - \frac{2+d}{2d(2(ad^2 + 2ad + a - d - 1) + d + 2)} \\ &= \frac{1}{d(d+1)} - \frac{2+d}{2d(2ad^2 + 4ad + 2a - 2d - 2 + d + 2)} \\ &= \frac{1}{d(d+1)} - \frac{2+d}{2d(2ad^2 + (4a-1)d + 2a)} \\ &= \frac{2d(2ad^2 + (4a-1)d + 2a) - d(d+1)(2+d)}{d(d+1) \cdot 2d(2ad^2 + (4a-1)d + 2a)} \end{aligned}$$

Because $4a-1 \geq 4 \cdot 3/8 - 1 = 1/2 \geq 0$, the denominator only has positive factors, which is why it is positive too. Therefore, it remains to show that the numerator

$$\begin{aligned}
&2d(2ad^2 + (4a-1)d + 2a) - d(d+1)(d+2) \\
&= 4ad^3 + (8a-2)d^2 + 4ad - d^3 - 2d^2 - d^2 - 2d \\
&= (4a-1)d^3 + (8a-5)d^2 + (4a-2)d \\
&= d\Big((4a-1)d^2 + (8a-5)d + (4a-2)\Big).
\end{aligned}$$

is positive too. Because d is always positive, we only consider the inner polynomial $p(d)$ in the following. Since p is a quadratic polynomial with a positive leading coefficient, it is sufficient to show that $p(5)$ and $p'(5)$ are positive.

$$\begin{aligned}
p(5) =&(4a-1)5^2 + (8a-5)5 + (4a-2) \\
=&100a - 25 + 40a - 25 + 4a - 2 \\
=&144a - 52
\end{aligned}$$

Therefore $p(5)$ is greater than 0 if $a \geq 52/144 = 0.36\overline{1}$ which is true for all $a \geq 3/8$. The derivative of p is $p'(d) = (8a-2)a + (8a-5)$. If we substitute 5 for a, we get

$$\begin{aligned}
0 \overset{!}{\leq} p'(5) &= (8a-2) \cdot 5 + (8a-5) \\
&= 40a - 10 + 8a - 5 = 48a - 15.
\end{aligned}$$

which is true for all $a \geq 15/48 = 0.3125$. Therefore $p'(d)$ is in particular positive for all $a \geq 3/8$. As a result Π_{d+1} is at least $1/(d+1) + 1/2$, which concludes the proof for the induction step. ☐

Using this lower bound for Π_d, we get $d \geq 1/2(n^2-n)$ from the inequality $N(n) \leq (d+2)/\Pi_d$. Consequently, the upper bound, we get from this approach, is at least quadratic. With the help of the next Lemma, we show that d actually is quadratic.

Lemma 6.5. *For every $d \geq 4$ and $a \in [3/8, 1/2]$, $\Pi_d \geq 1/a(1/d + 1/2)$ applies.*

Proof. We prove this by showing that Π_d is always smaller than 1, and $1/a * (1/d + 1/2)$ is always greater than 1, for every $d \geq 4$ and fixed $a \in [3/8, 1/2]$. The first inequality can be shown by using the lower bound to a:

$$\frac{1}{a}\left(\frac{1}{d} + \frac{1}{2}\right) \overset{a \leq 0.5}{\geq} \underbrace{\frac{2}{d}}_{\geq 0} + 1 \geq 1$$

For the second inequality, we show that every factor $(*)$ of Π_d is smaller than 1 but greater as 0, which is sufficient to show that Π_d itself is always smaller than 1. At first, we show that denominator of each factor is at least 1 and therefore positive.

$$\begin{aligned}
&1 \leq 2 \cdot f(i) + (i+1) = 2ai^2 + 3i + 1 \\
\Leftrightarrow\ & i^2 \geq -\frac{3i}{2a}
\end{aligned}$$

This is true for every $i \geq 0$ and $a \in [3/8, 1/2]$ because the left side is always greater than 0 and the left side is always smaller than 0. As a result, each factor can be bounded from above and below by

$$0 = 1 - 1 \overset{(*)\geq 1}{\leq} 1 - \frac{1}{2 \cdot f(i) + (i+1)} \overset{(*)\geq 0}{\leq} 1.$$

Since all factors of Π_d are greater than 0 and smaller than 1, Π_d is too. Consequently, $\Pi_d \geq 1 \geq 1/a(1/d + 1/2)$ applies. □

Using this inequality we get $d \leq 2a(n^2 - n)$ from the inequality $N(n) \leq (d+2)/\Pi_d$. As a result, we get an upper bound to the result of the MinDegree heuristic on an $n \times n$-grid, that is recorded in the following Theorem 6.6.

Theorem 6.6. *The* MinDegree *heuristic returns at most* $3/4(n^2 - n)$ *for an* $n \times n$*-grid.*

Proof. Substituting the lowest a, i.e., $a = 3/8$, into $d \leq 2a(n^2 - n)$ results in the desired upper bound for d. □

One could improve this result by finding a better function $f(d)$, but as long as $f(d) \in \mathcal{O}(d^2)$ this method still yields a quadratic upper bound.

6.3 Structural Investigation

In this section, we focus on the structure of possible elimination graphs of large grids under the MinDegree heuristic. We first point out the problem that arises in grids with sides length higher than 3, and give an alternative view on elimination graphs by introducing *eliminated components* in Section 6.3.1. We then find a way to illustrate the compressed information of eliminated components by introducing the *border graph* of an elimination graph in Section 6.3.2. Using this type of illustration, we finally describe the behavior of the MinDegree heuristic on large grids in Section 6.3.3.

6.3.1 Eliminated Components

The reason why we only consider grids with side lengths larger than 3 is, that in these and larger grids there is the possibility to eliminate an *inner vertex*, i.e., a vertex with degree 4. For all smaller grids, we figured out, that they are dismantled from the outside somehow. By eliminations of inner vertices, the neighborhood of *outer vertices* gets increasingly unpredictable. Additionally, Figure 6.1 shows that the whole grid can lose its uniform structure completely in the course of the MinDegree heuristic.

As a solution to this problem, we introduce a way to describe the structure of such elimination graphs properly and simpler by reducing the complexity of their cliques. It can be seen in Figure 6.1 that every clique only becomes bigger or merges with other neighboring cliques in the course of the heuristic. This idea leads to the following Lemma 6.7, which shows a connection between arbitrary graphs and the cliques of their elimination graphs.

Lemma 6.7. *Let* $G = (V, E)$ *be a graph,* $A \subseteq V$ *a set of vertices, and let* $G' = (V', E')$ *be the graph* G *after all vertices in* A *are eliminated. Further, let* $x, y \in V \setminus A$ *be two vertices that are not eliminated. Then the following equivalence applies:* x *and* y *are adjacent in* G' *if and only if there is a path* $P = (x, a_1, \ldots, a_p, y)$ *in* G *with* $a_1, \ldots, a_p \in A$ *and* $p \geq 0$.

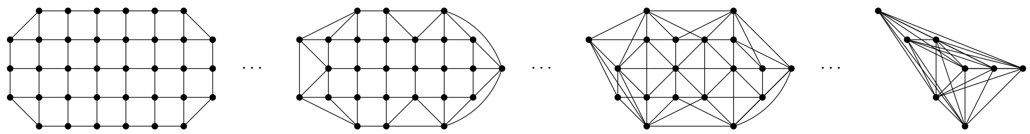

Figure 6.1: An example of the elimination of a 5×8-grid grid with random vertex ordering under the MinDegree heuristic. The graphs show the state of the grid after all 3-eliminations, 4-eliminations and 5-eliminations (from left to right). The last graph is the first complete graph in the series.

Proof. We first show the backward direction. We do this by proving the following statement: If all vertices $C \subseteq V$ of a connected subgraph of G are eliminated, then all vertices in the united neighborhood $N_G(C)$ are adjacent in the elimination graph. We prove this by induction over the size of C. In the base case $|C|$ is either 0 or 1. If $|C| = 0$, then $N_G(C)$ is empty and therefore a clique with size 0. If $|C| = 1$, the neighborhood of the one vertex $v \in C$ becomes a clique by the fill-in-operation during the elimination of v. In the induction step let $v \in C$ be a vertex and $w \in N_G(v)$ a neighbor of v. This vertex w exists because the subgraph induced by C is connected and $|C| \geq 2$. If we eliminate v from G, then all neighbors of v become neighbors of w in the elimination graph of G and v. By the induction hypothesis, all vertices in the new neighborhood of w will be a clique after all vertices in $C \setminus \{w\}$ are eliminated. Since the neighbors of v are a subset of those vertices the induction step is done. With this statement, we can show the backward direction. Let $A' \subseteq A$ be the set of vertices that are used in a path P between x and y. If we eliminate all vertices along this path, we know that the united neighborhood of these vertices will form a clique afterward. Since x and y are in this neighborhood, they are connected in G'.

For the other direction, we assume there is no such path P in G. Then x and y are not adjacent in G and there is an x, y-separator $S \subseteq V \setminus A$. This separator also exists in G', because no vertex in the separator is eliminated. Therefore x and y are not adjacent in G'. ☐

So, if we consider the components in the induced subgraph $G[A]$ of all eliminated vertices $A \subseteq V$, we can say something about the cliques in G'. Let $C \subseteq A$ be the vertices of a single component in $G[A]$, then the neighborhood $N_G(C)$ is a clique in G' according to Lemma 6.7, because for every pair of vertices $x, y \in N_G(C)$ there is a path $P = (x, c_1, \ldots, c_p, y)$ with $c_1, \ldots, c_p \in A$ and $p \geq 0$ in G. Additionally for every edge $(x, y) \in E' \setminus E$ there is such a component that has x and y in its neighborhood. We call such components C *eliminated components* of G with respect to A. The *order* of an eliminated component C is the size of its neighborhood $N_G(C)$.

6.3.2 The Border Graph

Figure 6.2 shows an example of how the eliminated components of a grid can be illustrated based on the area its neighborhood takes up.

The advantage of the right illustration in Figure 6.2 is that we can omit many interfering edges of the cliques, which makes the elimination graph structurally simpler. Additionally, no information is lost, since the size of the clique is represented by the number of vertices lying on the border of the areas. We call this kind of graph the *border graph* of an elimination graph. In the following, we show that the border graph of every elimination

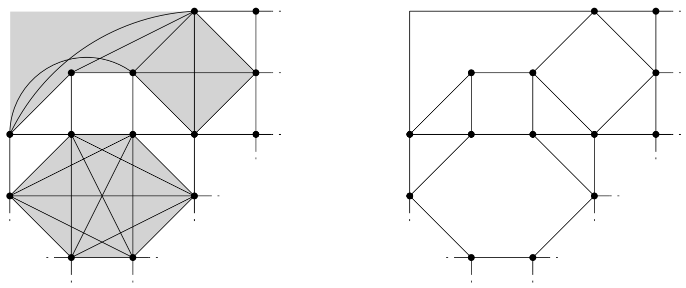

Figure 6.2: One possible elimination graph of a grid, where its the areas of eliminated components are colored in gray (left) or only displayed by their borders (right).

graph of a grid is planar by describing the construction of a planar embedding based on the uniform planar embedding of a grid. For this, we introduce a method to simulate eliminations that only uses edge contraction and the structure defined in the following Definition 6.8.

Definition 6.8. A *clique graph* of a graph $G = (V, E)$ is a bipartite graph $G_{\circledast} = (V, K, F)$ such that:

1. each $k \in K$ represents a clique in G. Those are called *clique vertices*, while all $v \in V$ are just called vertices.
2. an edge $\{v, k\} \in F$ exists if and only if the vertex $v \in V$ is part of the clique $k \in K$
3. for every $\{x, y\} \in E$ there is a $k \in K$ such that $\{x, k\}, \{y, k\} \in F$

Let $G = (V, E)$ be a graph. The method starts with the construction of a clique graph of G by converting every edge $\{u, w\} \in E$ of the original graph into a clique vertex, that is connected to u and w. Then every elimination of a vertex $v \in V$ is simulated in the perspective of clique graphs by the contraction of all edges that are incident to v in the clique graph. Lemma 6.9 shows that the so constructed graph is in fact a clique graph of the matching elimination graph of G. After each step, the original elimination graph can be constructed from the clique graph by eliminating all clique vertices (Lemma 6.10). A sketch of this process is given by Figure 6.3.

$$\begin{tikzcd} G \arrow[r] \arrow[d, Rightarrow] & G_{\circledast} \arrow[d, "6.9"] \\ G^A & G_{\circledast}^A \arrow[l, "6.10"] \end{tikzcd}$$

Figure 6.3: Process to simulate the elimination of a set vertices $A \subseteq V$ from a graph $G = (V, E)$ in the perspective of clique graphs. The graph G^A is the elimination graph of G with respect to A. The bold arrow indicates the actual elimination procedure, while the thin arrows describe the detour via the clique graphs. The graph $G_{\circledast}$ is obtained by the subdivision of all edges in G. The graph $G_{\circledast}^A$ is graph the graph $G_{\circledast}$ after the simulation of all eliminations of A.

In Lemma 6.9, we show that, given a clique graph $G_{\circledast}$ of G, the simulation step actually results in a clique graph of G's elimination graph. We prove this property for one vertex, which is sufficient to show the same statement inductive for a whole subset of vertices.

Lemma 6.9. *Let $G = (V, E)$ be a graph, $v \in V$ a vertex, G' their elimination graph and $G_{\circledast} = (V, K, F)$ a clique graph of G. Further, let $G'_{\circledast}$ be the graph $G_{\circledast}$ after all incident edge of v were contracted to a new clique k'. Then $G'_{\circledast}$ is a clique graph of G'.*

Proof. For this proof, we show that $G'_{\circledast}$ is still bipartite and that the neighborhood of every k in $G'_{\circledast}$ induces a clique in G'. We show the first property with the help of Figure 6.4. Before the contraction, all edges are incident to exactly one vertex $v \in V$ and one clique $k \in K$. After the contraction, all vertices on the second cycle are adjacent to v. Since v is now transformed to a clique vertex, and no other edges were changed, the graph remains bipartite.

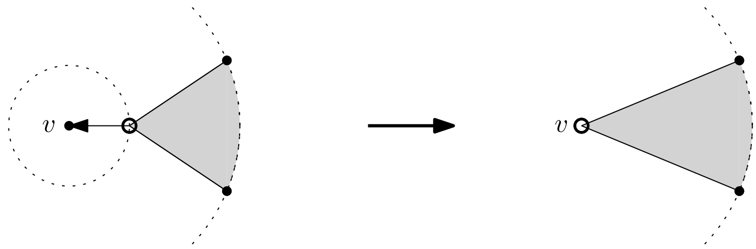

Figure 6.4: Before (left) and after (right) the elimination of v in G in the perspective of $G_{\circledast}$. Left: The vertex v is surrounded by clique vertices represented by a single $\circ$ and the dotted circle through it. This clique vertex is connected to several more vertices represented by the gray cone. Right: After the elimination of v (in G), v becomes a clique vertex in $G'_{\circledast}$.

To show the second property, let x and y be two vertices in $G'_{\circledast}$ that are adjacent to the clique vertex v. We now show that x and y are adjacent in G'. At first we show that x and y are adjacent to v in G. By construction of the clique vertex v, we know that there are clique vertices $k_x, k_y \in K$ such that x (or y) and v were connected over k_x (or k_y) in $G_{\circledast}$. Therefore there are cliques in G that included v and x (or y), which is why they were adjacent in G. Now, because the elimination of v creates a clique out of all its neighbors, x and y are adjacent in the elimination graph G'. $\square$

It remains to show that the original elimination graph G_A can be constructed from the clique graph $G^A_{\circledast}$ by eliminating all clique vertices. This is done in Lemma 6.10.

Lemma 6.10. *Let $G = (V, E)$ be a graph and $G_{\circledast} = (V, K, F)$ a clique graph of G. Then G is the elimination graph of $G_{\circledast}$ with respect to K.*

Proof. We show that the elimination graph of $G_{\circledast}$ with respect to K has the same set of vertices and the same set of edges as G. The sets of vertices match because, after the elimination of all clique vertices in $G_{\circledast}$ only the vertices $V' := (V \cup K) \setminus K = V$ remain. Now, let E' be the set of vertices of $G_{\circledast}$ after the elimination of all clique vertices. We show that $E = E'$, by showing both inclusions. Let $\{x, y\} \in E$ be an arbitrary edge in G, then there are edges $\{x, k\}, \{y, k\} \in F$ for some $k \in K$ in $G_{\circledast}$ by property 3. As neighbors of an eliminated vertex x and y are adjacent in the elimination graph of $G_{\circledast}$ with respect to K, so $E \subseteq E'$ applies. Now, let $\{u, v\} \in E'$. Since $G_{\circledast}$ is bipartite, and u and v are members of $V' = V$, there has to be a clique vertex $k \in K$ in $G_{\circledast}$ that is adjacent to both u and v. As $G_{\circledast}$ is a clique graph of G, both u and v are part of the same clique in G by property 2 and therefore adjacent in G. It follows that $E = E'$. $\square$

After showing how to simulate eliminations in the perspective of clique graphs and how to construct the original elimination graph back, we finally give a formal definition of the border graph and show a way to construct a planar embedding of it from the clique graph of an elimination graph, as depicted in Figure 6.5.

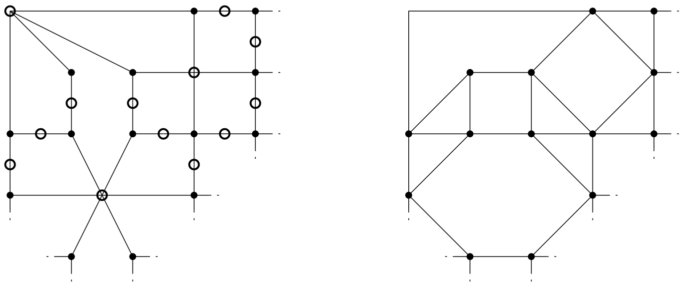

Figure 6.5: A clique graph (left) and the border graph (right).

For this, note that all clique graphs, used in this method are planar, if the input graph G is itself planar, e.g., a grid. The creation of the clique graph $G_{\circledast}$ from G is a subdivision of all edges, i.e., an operation that preserves the planarity of G. Moreover, every elimination in the perspective of clique graphs is a sequence of edge contractions, which also preserves the planarity. As a result, the basis for the definition of the border graph and the construction of its planar embedding is a planar graph.

To define the border graph properly, we first define the neighborhood area of a vertex on planar graphs. Let $G = (V, E)$ be a graph with planar embedding and $v \in V$ a vertex in G. Because G is planar, the neighbors of v can be ordered according to the angle of their outgoing edges of v. Let $w_1, \ldots, w_l$ be these vertices in the mentioned order. Now the *neighborhood area* of v is defined by the area the polygon formed by the lines between w_l and w_1 as well as w_i and w_{i+1} for all $i \in [1; l-1]$. The rhombus in Figure 6.5 for example is the neighborhood area of the vertex that once was in the middle of the rhombus. Finally, the *border graph* of an elimination graph G' of G is constructed by, first, constructing the planar clique graph described in Figure 6.3 and then replacing every clique vertex k (and its adjacent edges) by the edges that form the border of its neighborhood area. These edges exist because of Lemma 6.10. It remains to argue, that this embedding of a border graph is planar for every elimination graph of a grid. The reason for that is the property of the uniform grid embedding, that for every non-adjacent vertices x and y the neighborhood areas do not intersect. This way there can not be intersecting edges in the border graphs.

6.3.3 Observations on Large Grids

Using the border graph, we now have a simpler way to illustrate elimination graphs, which makes the observation of large cliques easier. In this subsection, we record some observations on how the MinDegree eliminates a grid and why it works the way it works. We motivate our observations based on the sequence of elimination graphs of an exemplary 10×16-grid grid given in Figure 6.6. The elimination graphs were generated by our program[1].

To discuss the behavior of the MinDegree heuristic on large grids for a small minimum degree, we introduce the concept of degree-independent vertices. For that let $G = (V, E)$ be a graph and let $u, v \in V$ two vertices. We say that u is *degree-independent (under elimination)* from v, if the elimination of v doe not change the degree of u. In Lemma 6.11 we give a characterization for degree-independent vertices.

[1] `https://github.com/Wombyte/thesis_tw_heuristics`

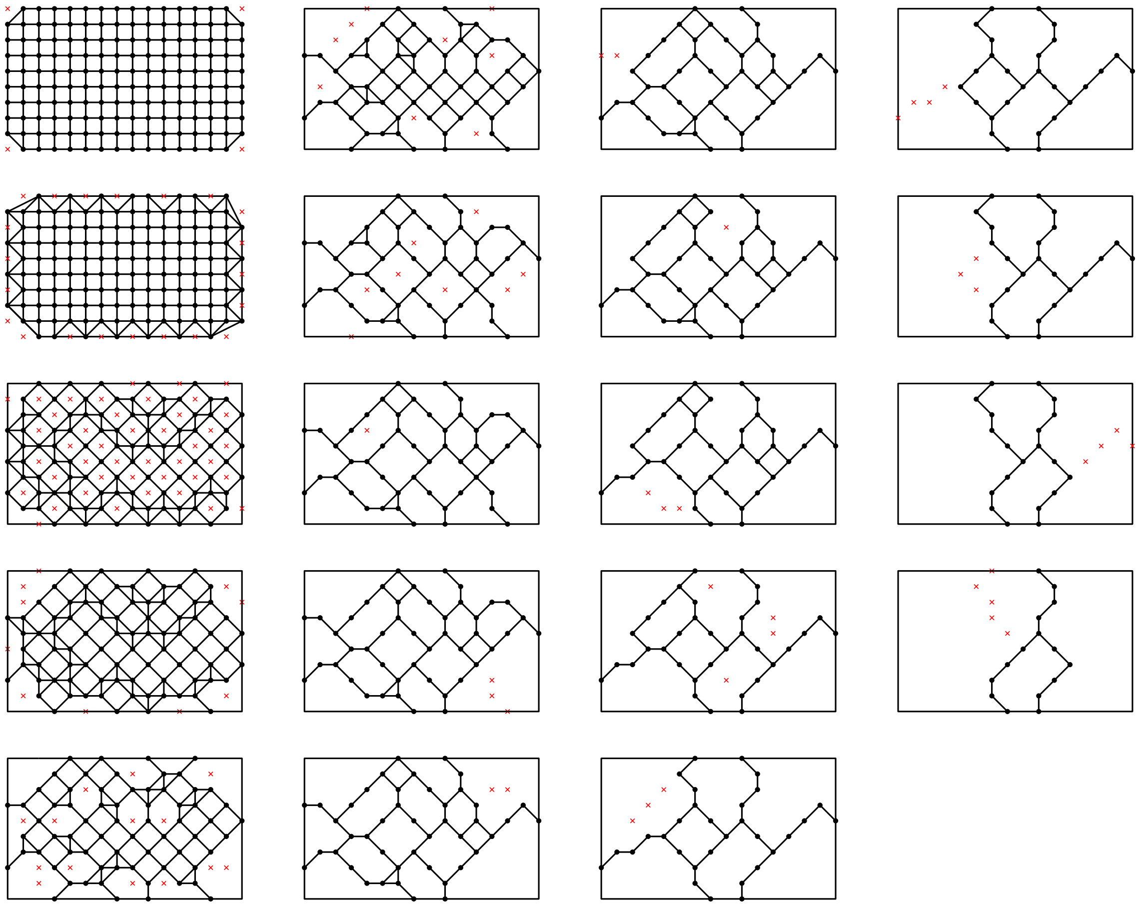

Figure 6.6: A sequence of border graphs of a 10×16-grid grid with random vertex ordering created by the MinDegree heuristic. For the sake of clarity, only elimination graphs are shown that are important for the following explanation. The red crosses mark the vertices that were eliminated since the last elimination graph.

Lemma 6.11. *Let $G = (V, E)$ be a graph with vertices $u, v \in V$. Then u is degree-independent from v if and only if $\{u, v\} \notin E$, or $\{u, v\} \in E$ and $|(N(v)\backslash u)\backslash(N(u)\backslash\{v\})| = 1$.*

Proof. If u and v are not adjacent, then u is not involved in the elimination of v, so its degree cannot be changed. Otherwise, let u and v be adjacent such that v has exactly one vertex w in its neighborhood that is not in the neighborhood of u. In this case, the degree of u stays the same because one edge, namely $\{u, v\}$ is removed, and one edge, namely $\{u, w\}$ is added in the course of the elimination of v.

For the other direction, we know that the elimination of v does not change the degree of u, i.e., the number of removed and added edges that are incident to u is the same. In the following, we make a case distinction over the number of removed edges incident to u. This value can only be zero if u and v are not adjacent, or one otherwise. In the case that u and v are adjacent, exactly one edge, namely $\{u, v\}$ is removed during the elimination. To keep its degree, u has to get one additional edge during the elimination of v. Therefore there is exactly one vertex in the neighborhood of v (besides u itself) that is not in the neighborhood of u. □

As a corollary, we can say that the minimum degree is not changed by one step of the MinDegree heuristic, if there are vertices u and v with minimum degree such that u

is degree-independent from v. Based on that we can understand the first steps of the MinDegree heuristic on large grids. The minimum degree of a large grid is 2 due to the vertices in the four corners of the grid. Since they are pairwise degree-independent each of them has to be eliminated before the minimum degree rises to three. The reason why exactly those four vertices are eliminated before the minimum degree increases is that the elimination of one corner vertex does not create a new vertex with degree 2.

The elimination graph after all corner vertices are eliminated is shown in the upper left corner of Figure 6.6. It has a minimum degree of 3, and all vertices on the border of the grid have this degree. Note that only the four pairs of vertices that were adjacent to a corner vertex are degree independent among the vertices with minimum degree. Therefore the four sides are independent in the sense that an elimination on one of the four sides does not affect the degree of a vertex on another side. So consider one of the sides with length n. Since every elimination removes one vertex and increases the degree of the two adjacent vertices (on the rim), at least $\lceil n/3 \rceil$ are eliminated on this side. Additionally, at most $\lceil n/2 \rceil$ vertices can be eliminated on such a side. The left side on the second elimination graph in Figure 6.6 shows an example of that.

As soon as the minimum degree of an elimination graph is 4, the behavior of the MinDegree heuristic becomes much harder to predict, since every set of pairwise degree-independent vertices can be eliminated due to the random vertex ordering. After such a set is eliminated the minimum degree increases to 5 (for sufficiently large grids), which means that for every vertex at least one of its neighbors is eliminated as the maximum degree in the original grid was 4.

To talk about the 5-eliminations, we divide the remaining vertices into two groups according to their horizontal and vertical neighbors. A vertex v is a horizontal (vertical) neighbor of a vertex u, if they are adjacent and lie on the same horizontal (vertical) line in the uniform embedding. It is easy to see that each vertex is adjacent to its horizontal and vertical neighbors in each elimination graph of a grid. We call a vertex with two horizontal and two vertical neighbors an *enclosed* vertex. Now consider an enclosed vertex v. Because the minimum degree is five, we know that at least one vertex u in the neighborhood of v already is eliminated. If u is an enclosed vertex itself, it creates a 4-clique during its elimination, which means that v got three new neighbors. Therefore no such vertex is eliminated until the minimum degree is lower than 6. As a result, only non-enclosed vertices are eliminated until then. This is shown in the fourth graph of Figure 6.6. The enclosed vertices that are part of one 4-clique can be eliminated afterward when the minimum degree is 6.

With the beginning of the second column in Figure 6.6 the eliminated components are clearly visible. At this point, the maximal cliques of the elimination graphs are mostly non-trivial, which is why we can reformulate the rule according to which the MinDegree heuristic chooses the next vertex. In the perspective of eliminated components, the MinDegree heuristic chooses a vertex whose neighboring eliminated components have as few vertices as possible in total and as many as possible vertices in common. At first, these vertices can be part of three or even four eliminated components, but with increasing iteration number it is more likely that an eliminated vertex was only part of at most two eliminated components. This applies to all eliminations after the second graph in the second column of Figure 6.6.

The bigger the eliminated components become the larger become their common *borders*, i.e., the vertices they share. If one of those vertices is eliminated the whole border is eliminated afterward. Therefore those steps are skipped in Figure 6.6, where this behavior can easily be seen in the last column. To give a formal reason for that let W be a border of the eliminated components C_1 and C_2, and let $v \in W$ a vertex with minimum degree. In fact, we know that all vertices that only belong to C_1 and C_2 have the same degree, i.e., the minimum degree. If we eliminate v, the degree of all those vertices is decreased by one because they had the same neighborhood as v in the previous elimination graph. Therefore they all take the new minimum degree and are consequently eliminated next.

We know from Theorem 6.3 that the size of the first complete elimination graph is the result of the MinDegree heuristic. In the viewpoint of eliminated components, one could think that this is only the case if there is only one eliminated component left, but actually, this may be the case sooner. The last border graph of Figure 6.6 for example belongs to a complete clique because for every vertex pair there is at least one eliminated component that contains both vertices of the pair.

In this section we have introduced the viewpoint of eliminated components, to bring some order in the seemingly chaotic structure of elimination graphs. We have used the border graphs to visualize these eliminated components in a clear way. During the definition of border graphs, we have introduced the concept of clique graphs and presented a method to simulate elimination on them only by edge contractions. With the help of the clique graphs, we have constructed planar embeddings for border graphs of grid eliminations-graphs, and thus showed that these border graphs are planar. Using their planar embeddings, we have described how the MinDegree heuristic behaves on large grids. We think that with the help of these findings, especially the concept of eliminated components, one can find a better upper bound than we found in Section 6.2.

7. Conclusion & Outlook

In this thesis, we have shown that the MinDegree heuristic is not only faster than the MinFillIn heuristic, but that there are also graphs on which the MinDegree heuristic achieves better results than the MinFillIn heuristic. For that, we have introduced a concept, that allowed us to transfer the behavior of the heuristics from one single graph to a whole family with arbitrary treewidth. We have thereby discovered that there is a series-parallel graph and a vertex ordering such that the MinFillIn heuristic, in contrast to the MinDegree heuristic, does not return the correct treewidth, while both heuristics always work optimally on chordal graphs and grids whose smaller side consists out of at most three vertices. During our investigations of the MinDegree heuristic on large grids, we have introduced a general technique to upper bound the heuristic's result by gradually increasing the minimum degree by eliminations with maximal fill-in. We have used this technique to find a quadratic upper bound to the heuristic's result on quadratic grids with respect to their side length. We have also presented new ideas to think about eliminations on grids and general graphs based on the idea of eliminated components. With the help of border graphs, we have visualized the eliminated components and, thus, described the behavior of MinDegree heuristic on grids in a clearer way. While proving the planarity of the border graphs of grids, we have introduced a method to simulate eliminations by edge contractions in the perspective of cliques graphs. We have, additionally, showed that these clique graphs are planar if the input graph is itself planar.

Since planar graphs have nice algorithmic properties, it would be interesting to see if this purely theoretical definition of clique graphs can be used to implement a faster version of the MinDegree heuristic for planar graphs. The problem that remains to be solved is how to chose the next vertex in the clique graph since the degree of a vertex in this graph does not necessarily correspond to the actual degree. If this problem is solved also the MinFillIn heuristic could be implemented fast. Besides that, also some of the other ideas presented in Section 6.3.3 could be used algorithmically to speed up the heuristic at least for some graph classes, if not for all. For example, the gradual elimination of a border with more than one vertex could be shortened.

Another direction for future work can be the improvement of the upper bound we have found in Section 6.2. A linear fill-in function $f(d)$, that could arise from considering the fill-in of eliminations with the same degree amortized, can result in a linear upper bound for the MinDegree heuristic on quadratic grids. But even finding a smaller quadratic factor for f could decrease the constant in the quadratic upper bound. However, we conjecture, based on the findings in Section 6.3, that there is an upper bound for the result of the MinDegree heuristic on an $n \times m$-grid that is linear in $n + m$ or even in $\min\{n, m\}$. Although we have not studied it much, a similar upper bound for the MinFillIn heuristic seems to be possible.

In general, we have mostly looked at grids in this thesis and only made a small detour to the other structures presented in the proof of the excluded grid theorem. Once better results for grids have been established, one could investigate if these results can be transferred to the more general structure beyond the grill. This would get us closer to answering the question of how good the heuristics are on general graphs. Of course, this target can also be reached from another starting point on, i.e., the chordal graphs. We have shown that both heuristics return the correct treewidth for all chordal graphs, but what about weaker forms of chordal graphs? Unfortunately, only weakening the definition of chordal graphs just by allowing larger chordless cycles will not bring the desired result, because in Section 3.1 we have already shown that the MinDegree heuristic can return an arbitrarily bad result on graphs with chordless cycles with length at most 5. Nevertheless, there are several other ways to extend the idea of chordal graphs that can be used.

Bibliography

[BB05] Emgad H. Bachoore and Hans L. Bodlaender. New upper bound heuristics for treewidth. In *Experimental and Efficient Algorithms (WEA)*, pages 216–227, 2005.

[BGK08] Hans L. Bodlaender, Alexander Grigoriev, and Arie M. C. A. Koster. Treewidth lower bounds with brambles. *Algorithmica*, 51(1):81–98, 2008.

[BHS03] Anne Berry, Pinar Heggernes, and Geneviève Simonet. The minimum degree heuristic and the minimal triangulation process. In *Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 58–70, 2003.

[BK10] Hans L. Bodlaender and Arie M. C. A. Koster. Treewidth computations i. upper bounds. *Information and Computation*, 208(3):259–275, 2010.

[Bod98] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[Bod05] Hans L. Bodlaender. Discovering treewidth. In *Theory and Practice of Computer Science (SOFSEM)*, volume 3381, pages 1–16, 2005.

[CCMN03] François Clautiaux, Jacques Carlier, Aziz Moukrim, and Stéphane Nègre. New lower and upper bounds for graph treewidth. In *Experimental and Efficient Algorithms (WEA)*, pages 70–80, 2003.

[Chl92] Janka Chlebikova. On the tree-width of a graph. *Acta Mathematica Universitatei Comenianae*, 61(2):225–236, 1992.

[DHJ+16] Holger Dell, Thore Husfeldt, Bart M. P. Jansen, Petteri Kaski, Christian Komusiewicz, and Frances A. Rosamond. The first parameterized algorithms and computational experiments challenge. In *International Symposium on Parameterized and Exact Computation (IPEC)*, pages 30:1–30:9, 2016.

[Die12] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[Epp11] David Eppstein. `https://cstheory.stackexchange.com/questions/9008/forbidden-minors-for-bounded-treewidth-graphs/`, 2011. [Online; accessed 20-August-2021].

[Jr.83] Leslie E. Trotter Jr. Algorithmic graph theory and perfect graphs. *Networks*, 13(2):304–305, 1983.

[Kur30] Kazimierz Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae*, 15(1):271–283, 1930.

[MSJ19a] Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In *International Conference on Database Theory (ICDT)*, pages 12:1–12:18, 2019.

[MSJ19b] Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data (extended version), 2019.

[Nor17] Sergey Norin. `https://www.math.mcgill.ca/snorin/math599w2017/Notes.pdf`, 2017. [Online; accessed 25-August-2021].

[PS97] Andreas Parra and Petra Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics*, 79(1-3):171–188, 1997.

[RS86a] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

[RS86b] Neil Robertson and Paul D. Seymour. Graph minors. v. excluding a planar graph. *Journal of Combinatorial Theory*, 41(1):92–114, 1986.

[RS04] Neil Robertson and Paul D. Seymour. Graph minors. XX. wagner's conjecture. *Journal of Combinatorial Theory*, 92(2):325–357, 2004.

[ST93] Paul D. Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory*, 58(1):22–33, 1993.

[vDvdHS06] Thomas van Dijk, Jan-Pieter van den Heuvel, and Wouter Slob. Computing treewidth with libtw, 2006.

[Wag37] Klaus Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.